A COMPUTER PROGRAM FOR THE ANALYSIS OF LINEARLY ELASTIC PLANE-STRESS, PLANE-STRAIN PROBLEMS

by

John Patrick Malone

UNITED STATES
NAVAL POSTGRADUATE SCHOOL

THESIS

A COMPUTER PROGRAM FOR THE ANALYSIS OF

LINEARLY ELASTIC PLANE-STRESS, PLANE-STRAIN PROBLEMS

by

John Patrick Malone

September 1968

A COMPUTER PROGRAM FOR THE ANALYSIS OF LINEARLY ELASTIC PLANE-STRESS, PLANE-STRAIN PROBLEMS

by

John Patrick Malone
Lieutenant, United States Navy
B.Me., The Ohio State University, 1960

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1968

ABSTRACT

The computer program for the analysis of linearly elastic plane-stress or plane-strain problems devised by Felippa in his work on "Refined Finite Element Analysis of Linear and Nonlinear Two-dimensional Structures" has been modified to include the use of initial displacement boundary conditions. In addition the original IBM 7094 computer dependent program has been adapted for use on the IBM 360/65 computer. In both programs the FORTRAN IV language has been used.

Problems involving "Poor fit" displacement boundary conditions and refined mesh analysis using coarse mesh analysis input displacements, which could not have been done with the original program, are now possible with the modified version presented herein.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

σ_x	Stress in X-Direction
σ_y	Stress in Y-Direction
τ_{xy}	Shearing Stress
σ_{max}	Maximum Principal Stress
σ_{min}	Minimum Principal Stress
τ_{max}	Maximum Shearing Stress
X, Y	Global Coordinate System and Coordinates
u	Displacement in X-Direction
v	Displacement in Y-Direction
E	Modulus of Elasticity
ν	Poisson's Ratio
γ	Specific Weight
α	Coefficient of Thermal Expansion
ϕ	Angle of line Boundary Condition
$\{F\}$	Nodal Point Force Vector
$\{\delta\}$	Nodal Point Displacement Vector
$[k]_e$	Element Stiffness Matrix
$[K]$	Complete Stiffness Matrix
$[B]$	Strain-Displacement Matrix
$[D]$	Stress-Strain Matrix
$[\ ]^T$	Matrix Transpose

ACKNOWLEDGEMENTS

The computer program described herein was revised and adapted during the author's graduate study for the M. S. degree in Mechanical Engineering at the Naval Postgraduate School.

The author wishes to express his gratitude to Dr. Carlos A. Felippa, currently at Boeing Aircraft, the originator of the program. He is also grateful to Professor Gilles Cantin of the Naval Postgraduate School for having supervised the work and also for his constant encouragements throughout.

The Naval Postgraduate School Computer Center provided facilities for the computer work.

Chapter I Introduction

In this chapter reasons are given for the use of computer structural analysis. The purpose of the presentation is explained, and some expected program uses are listed. The objectives and extent of work completed by the author are presented.

1.1 Reasons for computer analysis

The engineer of the present day is faced with solving structural problems of great complexity. Classical mathematics, despite its ever increasing sophistication, is only capable of solving severely idealized situations while at the same time placing a burden on skilled manpower which could be better used for design and development processes.

Fortunately, the simultaneous development of the digital computer and new general methods of solution for problems in Continuum Mechanics has come to the rescue in many areas of investigation. The modern digital computer coupled with the finite element method is revolutionizing the approach to the process of analysis. Expensive and time consuming experimental models, now often used in the design of important structures, are rapidly becoming displaced by more economical computation.

1.2 Purpose of presentation

This presentation offers a computer program which takes advantage of the modern computer's high computational speed and the versatility of the finite element method. The program is for the general analysis of arbitrary plane-stress or plane-strain structural mechanics problems in linear elasticity. Sufficient background material on both the finite element method and the program structure is given so that those not conversant with these matters may use the program effectively. In addition, information on the practical aspects of the program alogrithm for the direct stiffness procedures, boundary condition application, and solution of large systems of linear equations are given for those interested in making modifications or additions to the program.

1.3 Expected uses

Two-dimentional elastic analysis by the finite element method is well suited for the development of large scale, production usage computer programs. Expected areas of use for the program presented include:

(1) Practical working design situations, ranging from one-time preliminary analyses to extensive in-depth studies.

(2) Design data generation for stress concentration factors or design tables for specific structural shapes.

(3) Augmentation of other analysis methods. Localized area analysis. Utilization of the thermal stress and body force features. Gross analysis to determine instrument or sensor placement.

1.4 Scope and objectives

The original version of the program was written by Dr. Carlos A. Felippa [3]* during the course of his Ph.D. studies at the University of California. The original program was dependent on the IBM 7094 computer and associated hardware then at the Berkeley Computer Center.

The extent of the author's objectives and work on the program has been: (1) reprogramming modifications necessary for the adaptation of the program to the faster IBM 360/65 computer at the Naval Postgraduate School Computer Center, (2) substituting the use of the more expedient random access disk unit for the original tape storage methods which were used for

*Numbers in brackets refer to the list of references on page 59 portions of the external storage requirements, (3) restructuring and modifying the program to allow usage of initial displacement boundary conditions, thus allowing the program to accept all physical boundary conditions for planar problems, (4) creation of a "User's Manual" to facilitate use of the program by persons not familiar with the finite element method or production programming techniques. This presentation is orientated toward this final objective.

Chapter II General Program Features

In this brief chapter the general program capabilities and features are given as background information.

2.1 Capabilities

The program in its current form represents a large scale, high speed general computational processor for the stress analysis of plane elastic bodies. It is one of the few such programs available in the open literature.

The program is capable of analyzing any arbitrary plane structural shape, singly or multiply connected. The body thickness may vary, to a degree. All physical boundary conditions are acceptable. Structural loading types may include surface forces (concentrated and distributed), body forces, and thermal loading effects. Any linearly elastic, homogenous, isotropic material may be analysed. Up to six different materials are allowable for composite material structures.

2.2 Analytic results

The method of analysis is the finite element procedure [2], [16], with the displacement solution method and direct stiffness matrix generation [12], [13]. Displacement compatible finite elements are used. This type of element has the virtue in theory that if finer and finer mesh subdivisions are used and there is no numerical error or round off, convergence to the exact solution is assured [11].

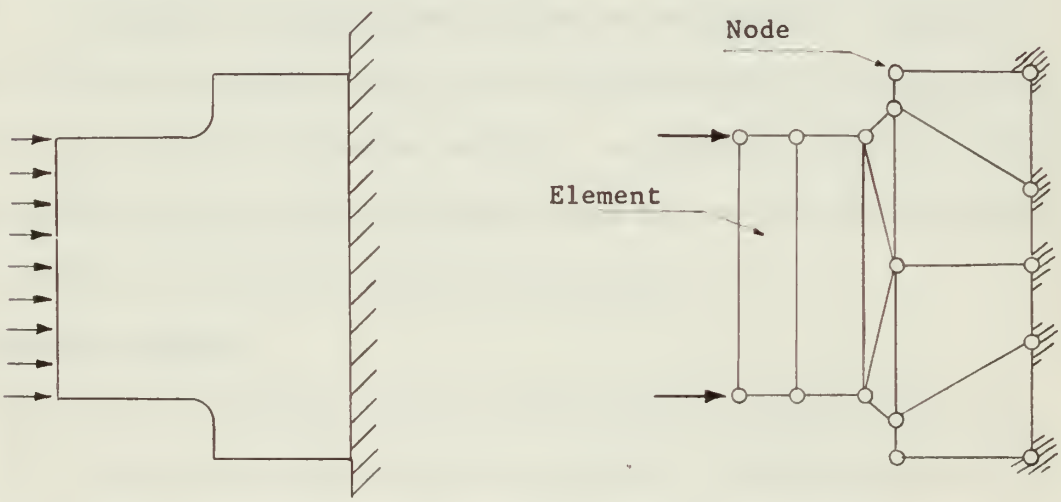
In the analysis the program computes in-plane deflections and stresses at selected sites on the body, resulting from in-plane loading. Stress components σ_x, σ_y, and τ_{xy} are computed, as well as the principal stresses and directions at every nodal point. Stress contour graphs of the structure, or a magnified subportion, are generated, wherein the structure outline and constant stress level contour lines are printed.

Chapter III Fundamentals of the Finite Element Method

In this chapter, the fundamentals of the finite element method are given in order to acquaint the user with the procedures and the nomenclature needed to understand the program structure and fully utilize its potentials.

3.1 Basic idealization

The basic concept of the plane finite element method is that any continuous two-dimensional body may be separated by imaginary lines into a finite number of individual elements, i.e., an assemblage of smaller individual plates. Thus an actual continuous structure is replaced by an assemblage of discrete structural elements (Fig. 3.1).



Actual structure and load

Idealized structure and equivalent loads

Figure 3.1 The Finite Element Idealization

The elements are assumed to be interconnected only at a discrete number of joints or nodal points through which forces are transmitted, but the structural elements are such that the connectivity of the structural system is preserved along all the common boundries of adjacent elements.

With these stipulations: (1) continuum approximation by an assemblage of elements and (2) element displacement field restrictions, we have, in effect, converted a continuous body into a body composed of many structural elements where each element can deform in only a certain number of predictable shapes.

A given point in the continuum has two independent degrees of freedom of displacement. The corresponding point in the idealized structure still has two degrees of freedom, but the displacement field is now restricted to the one selected and imposed on the element. This reduction is basic to any solution. In the finite element method the reduction is made in a physical manner using structural elements that can retain many of the physical properties and behavior characteristics of the original material. In other methods of analysis the reduction is often made in a mathematical procedure, in which many of the desirable properties and characteristics are lost through inability to make adequate or tractable mathematical models.

3.2 Displacement fields

The constraints we have placed on the element take the form of displacement functions, or fields, for which the overall element displacement shapes, and resulting strains and stresses, are functionally related to the nodal point displacements (generalized coordinates). The success of the element in duplicating the performance of the actual continuous body is critically dependent on the choice of the displacement function. Much effort has been given to the determination of successful displacement functions and the fundamental rules governing their generation. For flat elements, general procedures have been developed [3], however as yet there is no such general method for the construction of displacement fields for a curved element [1].

3.3 Convergence requirements

To guarantee that a finite element solution will converge to the true solution, under conditions of increasingly finer element division (finer mesh), the planar displacement function must be able to satisfy three basic conditions:

(1) Continuity condition: interelement displacement compatability must be satisfied. For plane cases this implies that adjacent element edges must remain together under element deformation.

(2) Completeness conditions: (a) rigid body modes - overall rigid body displacement of the element must not result in element straining; (b) constant strain states - nodal displacements that are indicative of constant strain must result in constant strain conditions in an element.

(3) Invariance - element stiffness properties derived through the use of displacement functions must remain the same for all coordinate systems which are used for their derivation.

3.4 Element equilibrium equations and stiffnesses

Knowing element displacement modes as functions of nodal displacements we can now write the equilibrium equations for the element. The relations between nodal displacements and nodal forces (generalized forces) are expressed in compact form by the element stiffness matrix. The resulting element equilibrium equations take the form:

$$\left\{F\right\}_e = \left[k\right]_e \left\{\delta\right\}_e \tag{3.1}$$

Where,

$\left\{F\right\}_e$ = Nodal point force vector for an element, (2N x 1)

$\left\{\delta\right\}_e$ = Nodal point displacement vector for an element, (2N x 1)

$[k]_e$ = Element stiffness matrix, symmetric, (2N x 2N). (N is the number of nodes per element)

The element stiffness matrix is a function of the geometric and physical properties of the element.

The derivation of element stiffnesses can be approached using the principle of virtual displacement [13], [16]. Here, virtual nodal displacements in a local frame of reference are imposed on the element. The external and internal work done by the various forces and stresses during displacement are equated. The resulting equation can be reduced to the general form:

$$\left\{F\right\}_e = \left[\int [B]^T [D] [B] \, d(\text{vol.})\right] \left\{\delta\right\}_e \tag{3.2}$$

where,

[B] = Strain-Displacement relationships from elasticity considerations.

[D] = Stress-Strain relationships from elementary theory.

The element stiffness is developed by performing the indicated matrix integration.

An alternate method of deriving the element stiffness is an energy approach [3], [5]. Here the total potential energy of the element-load system is minimized, an outgrowth of the application of the governing variational principle. This is, in fact, a form of the Ritz technique applied to the network of finite elements. This method is not as direct, nor as physically interpretable as the previous approach. However it is a more powerful technique and leads naturally to the formulation of the consistent mass matrix, consistent stiffness matrix and consistent load vectors. Elements having sophisticated geometries and higher order displacement fields can be handled practically only with the energy method.

Having the element stiffness matrix and the two auxiliary matrices [B] and [D] we can develop the full stress analysis for a single element when the nodal displacements resulting from the loading are known.

3.5 Complete stiffness matrix

To permit combining the individual stiffnesses each must first be transformed from its local coordinate system to a common (global) system for the entire node-element mesh. This is done with a conventional tensor transformation relating the two coordinate systems. The formulation of the complete stiffness matrix for the discretized structure can now be accomplished. In the direct stiffness method this is achieved by the direct addition of the element stiffnesses at all the element interface nodal points. This is a technique idealy suited for computer operation (see appendix 2).

3.6 System equilibrium equations

The complete stiffness matrix [K] is the matrix of equilibrium equations for the total idealized structure. The set of overall equilibrium equations is of the same form as equation (3.1), viz.:

$$\{F\} = [K]\{\delta\} \tag{3.3}$$

The nodal point force vector (load vector) contains the loads to which the body is subjected. The loads are in the form of concentrated nodal point forces directed along the global coordinates axes. Where the body loading is in the form of distributed forces, body force loads or thermal loads, equivalent concentrated forces are applied at the appropriate nodal points.

The nodal point displacement vector (displacement vector) is the unknown in equation (3.3). This is a direct result of the formulation developed and is responsible for the name, "Displacement Solution Method."

Another general method of finite analysis is available, where a stress field is used [5], this procedure is called the "Equilibrium Method."

The overall equilibrium equations must be modified by the boundary conditions of the problem. This involves elimination of stiffness contributions in [K] for constrained nodes. In the case of initial displacement boundary conditions the load vector and displacement vector must be adjusted to reflect equivalent nodal forces and the initial displacement value, respectively, that result from displacing a node a known amount.

The final set of equations (3.3) can attain rather large proportions. The order of the load and displacement vectors and the size of the symmetric overall stiffness matrix are 2 x P, where P is the total number of nodal points in the mesh. Fortunately the stiffness matrix is sparsely populated and banded, which permits specialized solution techniques on a computer. Actually, the displacement solution method and the direct stiffness procedures are used so that these stiffness matrix properties may be exploited (see appendix 2).

The final equations (3.3) are solved for the nodal point displacements. With the known displacements, the matrices [B] and [D] may be applied for each element and the overall nodal stresses evaluated. The stress levels at nodes common to more than one element are usually averaged to obtain a representative value. The end result is the displacements and stress levels at each node of the idealized structure.

Chapter IV Program Information and Structure

This chapter offers information on the program and gives its structure, functional routines and general method of operation.

4.1 Program identification

PSELST - Plane Stress Elastic Analysis using Linear Strain Triangles.

Programmed: Carlos A. Felippa, June 1966

Revised: John P. Malone, July 1968

4.2 Purpose

The purpose of the program is to provide a high speed, production use computational solution of general plane-stress or plane-strain static, linear elastic problems using linear strain triangles in the finite element method. Surface loads, body forces and thermal effects may be considered.

4.3 Programming information

The program is written in FORTRAN IV language [6] for the IBM OS/360 Model 67 computer. An overlay structure [7], [8], under control of the system linkage editor, is utilized to conserve main core storage. The overlay segments are arranged so that only active subroutines and attendant internal storage requirements are in main core at a time.

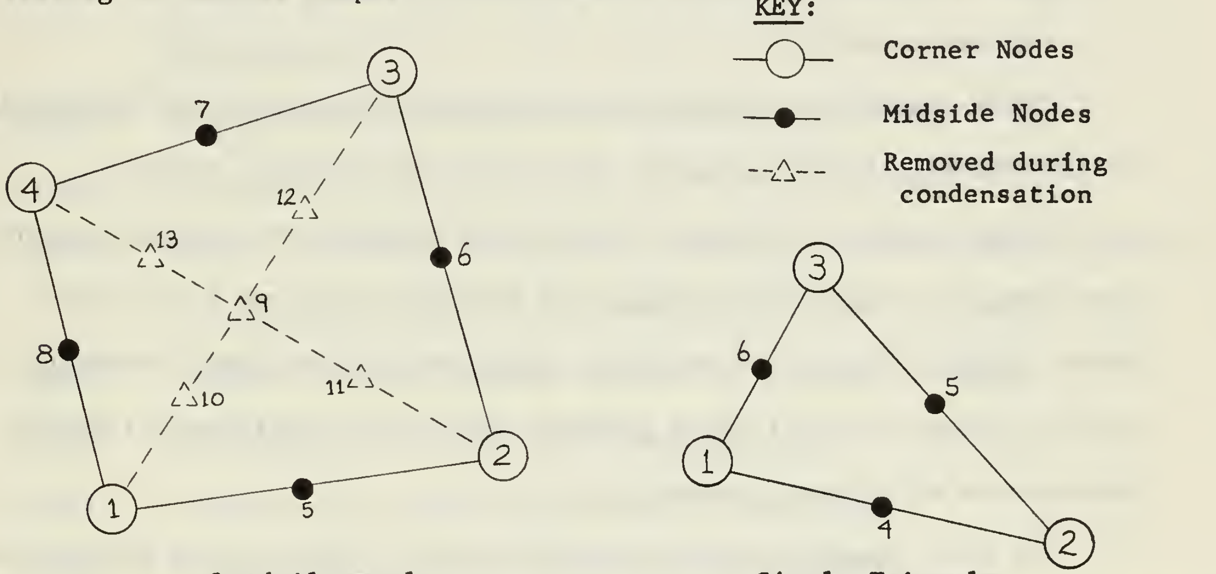
4.4 External storage

Fortran logical units 1,2,3,7,8, and 9 are used for temporary storage. Logical unit 7 is defined as a data set on one IBM 2311 random access disk storage unit. The remaining logical units are defined as separate data set blocks on a second 2311 disk unit. The input card reader, output printer, and output card punch are defined as units 5,6, and 4 respectively.

4.5 Basic finite element mesh units

The basic mesh element is a quadrilateral composed of four 6-nodal point linear strain triangles (LST), the center point being the centroid, (Fig. 4.1). Internal points 9 to 13 are eliminated by matrix condensation [3], [9] thereby reducing the number of degrees of freedom from 26 to 16.

Single 6-nodal point triangles may also be specified to facilitate fitting of certain shapes.



Quadrilateral
(4 corner nodes, 8 external nodes)
(#9 = centroid)

Single Triangle
(3 corner nodes, 6 external nodes)

Figure 4.1 Basic Finite Element Mesh Shapes

4.6 Capacity

The mesh input is subject to the following limitations for an IBM computer with 256 K bytes of main core storage;

Max. number of elements (Quad. and/or triangle) - - 350

Max. number of external (total) nodal points - - - 1050

Max. number of restrained components - - - - - - - 250

Max. difference of nodal point numbers for the same element - - - - - - - - - - - - - - - - - - - 79

These limits are dictated by storage requirements in the computation of the overall stiffness matrix, and not by the equation solver.

It is estimated that with 512 K bytes of main core storage the program could be modified to accept 1700 external nodes and a half-bandwidth of 250. This would result in approximately 600 usable elements. Solution times for full scale problems with these modifications may be prohibitive on current computers.

The most commonly encountered limitation of the current program version is the maximum nodal point number difference. Next in order are maximum external nodes (total nodes), maximum number of elements, and maximum number of restrained components.

The maximum number of degrees of freedom is 1050 x 2(external nodes) + 350 x 10(condensed and recovered nodes) = 5600. The condensation procedure reduces the number of equations to 2100, yet retains the versatility of 5600 degrees of freedom.

The maximum half-bandwidth (2 x Max. node number difference + 2), a measure of the width of the banded area of the overall stiffness matrix, is 160.

4.7 Program structure

The overlay root-segment structure is shown in Figure 4.2, where each rectangle represents a subroutine. The subroutine functions are:

MAIN	remains in core and controls calling sequence during execution;
RDDISK-WRDISK	remains in core and provides high-speed deposit and retrieval of data on a random access disk storage unit;
SETUP	inputs, prints and checks mesh data, and evaluates element stiffnesses;
STQUAD	assembles and condenses quadrilateral stiffness;
STLST6	computes stiffness of a six nodal point triangle;
LDINPT	inputs load cases and reduces surface, body, and thermal loads to equivalent forces on element external nodal points;
THERLD	computes initial thermal forces for a single triangle;
FORMK	assembles the complete stiffness matrix;
SOLVE	obtains nodal point displacements from BIGSOL;
BIGSOL	solves large capacity banded matrix problems;
STRESS	evaluates and prints element and nodal stresses;
TRISTR	computes stresses for a single triangle;
CNTPLT	produces printer plots of stress contour lines.

The subroutine flow chart is presented in Fig. 4.3.

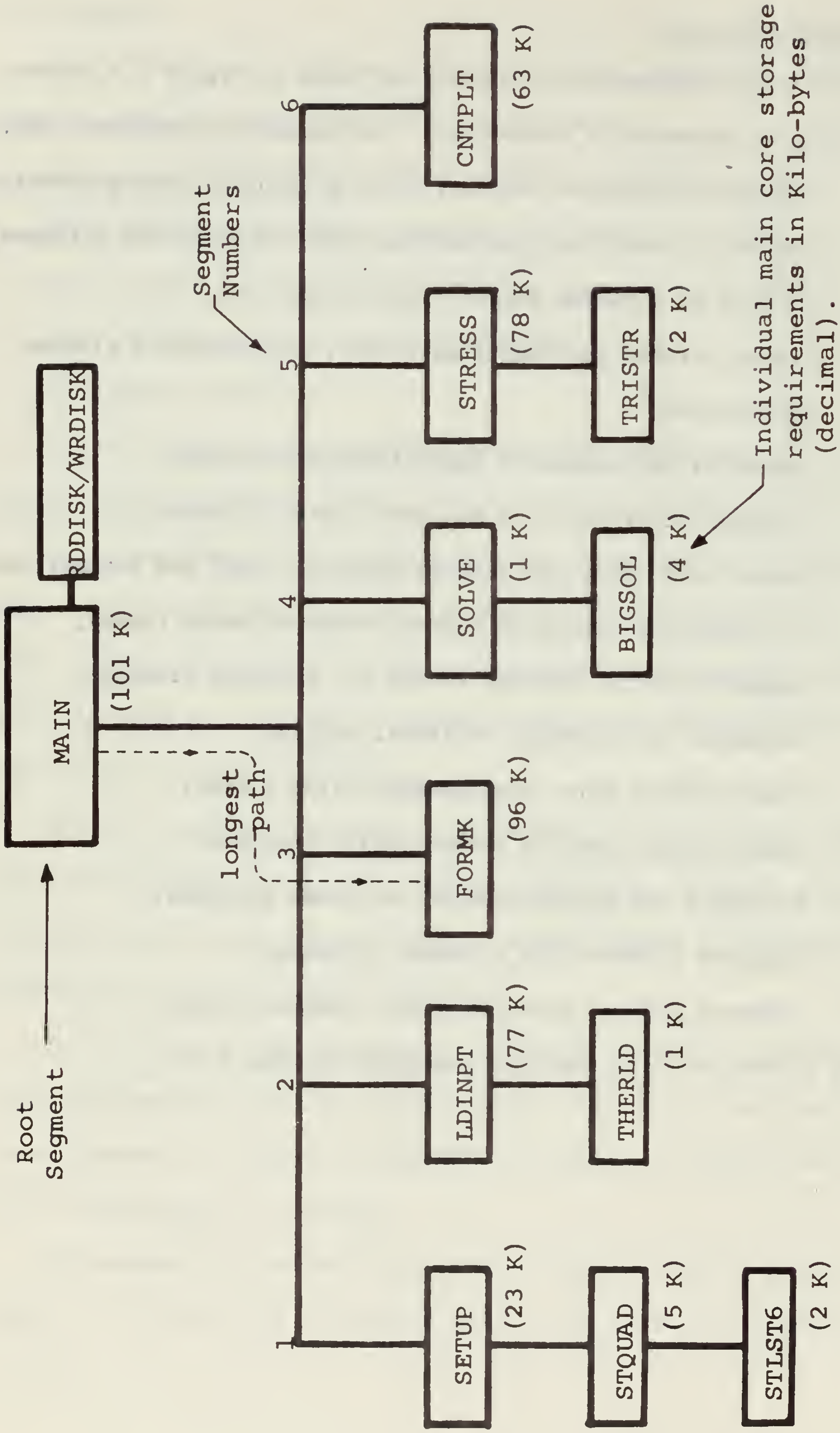


Figure 4.2 Program Overlay Structure

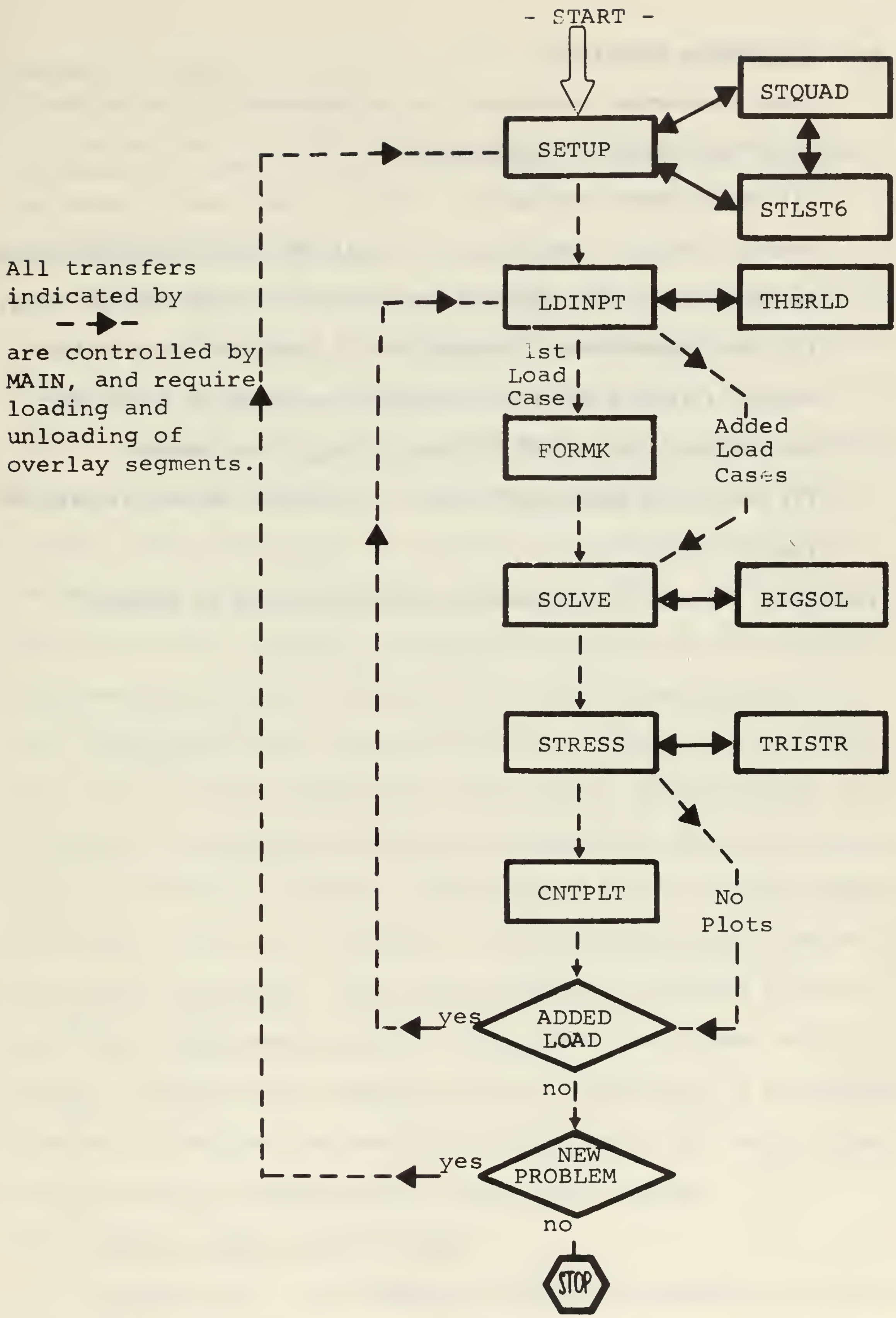


Figure 4.3 Subroutine Flow Chart

4.8 Programming techniques

Three programming techniques that are contained in the original version of the program are noteworthy:

(1) RDDISK/WRDISK Subroutine - used as a data transfer device to external storage. Benefits are: simplified data address accounting in subroutines, fast data transfer, access to large storage areas.

(2) Matrix Compression (condensation) - upper half-band of the overall stiffness matrix is stored, column-wise, by blocks for ease of manipulation and to reduce storage requirements.

(3) Use of one dimensional array - to minimize address calculation time.

Techniques (2) and (3) are described more completely in appendix 2.

Chapter V User Information

In this chapter specific features of the program are explored. Techniques are given to assist in the effective use of the program.

5.1 Typical applications

Representative applications of specific problem types are illustrated in Fig. 5.1. Case (a) is a wing frame where gust loading is simulated by a static load situation. Thermal loading effects at the frame tip might also be considered. Case (b) is a cross-section of a length of pressure piping. The circular perimeters have been distorted during a bending process. The stress levels and patterns at the thin wall section are desired. Case (c) is a comparison study of two welds. The left fillet welds are cheaper to produce in some given process, but the right side full-penetration welds are expected to be significantly stronger. A cost effectiveness study requires information comparing the two welds. Case (d) is a stress concentration factor study. Data for design tables is desired to guide design engineers confronted with this configuration. Loading situations of tension, compression and in-plane bending might be considered. Case (e) is a section of a prefabricated tunnel proposed for underwater installation. The tunnel is partially evacuated as part of a high speed transportation scheme. The optimum tunnel cross section is desired, using a minimum material criterion. Case (f) is a hypothetical plane-strain problem illustrating the versatility of the finite element technique used in conjunction with a high speed computer.

5.2 General program input and output

The versatility of the program is obtained by accepting a penalty of requiring a large amount of input data. The data is in the form of standard punched cards. The number of input cards will be between 25 and 500, dependent on the number of elements used and the extent of loading.

Input information is divided in two groups, structure data and loading data. The general content of input data follows:

1. Structure data.
   a. Accounting totals - number of elements, nodes, etc.
   b. Mesh configuration.
      1. Node-to-element identification.
      2. Corner nodes, X-Y coordinates.
   c. Material physical constants.
   d. Boundary conditions
2. Loading data.
   a. Accounting totals - number and types of loads.
   b. Load magnitudes and locations.

Once a problem has been set up with the structure data input, repeated load cases may be run. Consecutive problems may also be run.

Program output is divided into three parts: (1) echo check of input data, (2) displacements and stresses, (3) contour maps. The general content of each part follows:

1. Echo check.
   a. Input structure data printed for each problem.
   b. Input loading data printed for each load case.
2. Displacements and stresses.
   a. Final load vector.
   b. Nodal point displacements printed (punch option).
   c. Computed element nodal stresses (optional).
   d. Averaged nodal stresses, σ_x, σ_y, τ_{xy}, σ_{max}, σ_{min}, τ_{max}, and maximum principle stress orientation (punch option).

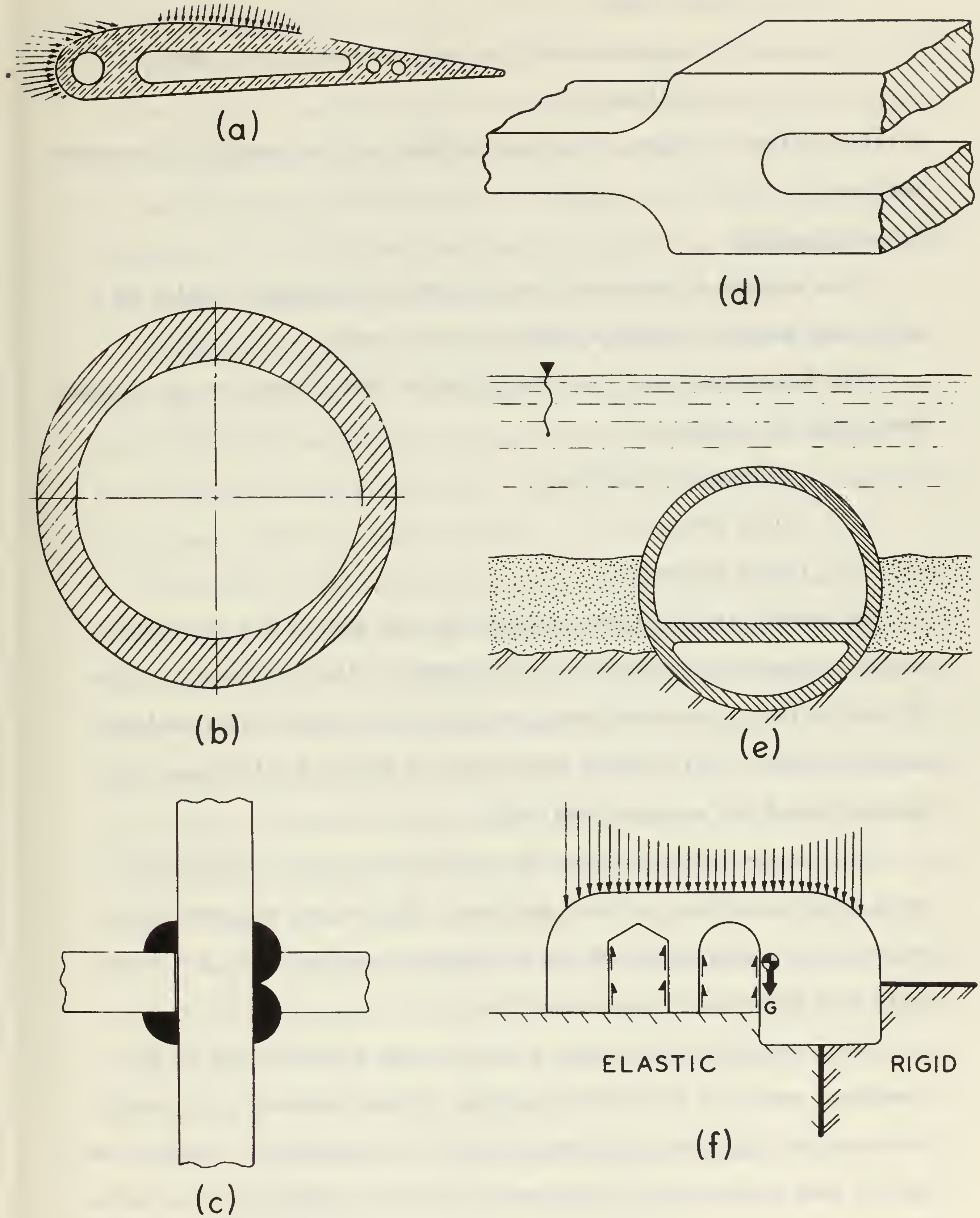


FIGURE 5.1 Typical Applications

3. Contour graphs

a. A contour graph for each averaged stress computed (optional).

Details of input, output and program options will be covered in subsequent sections.

5.3 Precautions

This section is presented here to prevent users from rushing to a card punch machine with this manual in their hand.

The author has found that the following "rules" must not be violated when using the program.

(1) Know your problem.

(2) Be patient.

(3) Be prepared.

An example of the first rule involves how much of the structure to analyze. Many structures have 1-fold symmetry, less frequently 2-fold. If the loading is symmetric about the same axes, partial body analysis may be utilized. This avoids duplication of data and allows more effective use of the elements available.

The rule on patience should be applied during mesh construction, nodal point numbering, and card punching. The program has many error printout exits that determine and illuminate input mistakes, but these exits also terminate program execution.

Being prepared means having a plan to make effective use of the tremendous amount of information that the program produces. A cursory glance at the sigma-max graph to determine if a material is suitable may fail to make adequate use of the wealth of output information available. Other persons may need information from your analysis or some unusual condition may exist that warrents further study.

5.4 Mesh construction

An effective mesh is one that represents a good approximation to the true structure. How effective a mesh is may not be known until one run has been completed and stress patterns are reviewed.

An efficient mesh is one that is formed on the basis of the above requirements, but also allows computation to proceed as rapidly as possible. This section deals with the construction of efficient meshes.

The mesh idealization of the body may contain 1 to 350 basic mesh units (Fig. 4.1). Due to its superior stiffness properties, use of the quadrilateral is preferred. The use of triangles should be limited to curve fitting difficulties and when changing the fineness (or coarseness) of the mesh; this is termed grading.

The nodes of the basic mesh unit are called corner points or midside points. Mesh units may only be joined with correspondance between corner and midside points. Two adjoining units share one midside point and two corner points. In a mesh, corner points will join three (usually four) or more elements when internal to the body (Fig. 5.2c, points a and b). Midside points are always common to only two elements when within the body interior. Each element and node must be numbered for identification purposes. Element size, shape and location information is determined by listing the X-Y (global) coordinates of all corner points. Midside point coordinates and element local coordinate systems are internally generated in the program. Nodal numbering techniques and coordinate system usage are covered in the following sections.

Three factors should be considered in constructing the mesh:

(1) Expected or known stress gradients.

(2) Mesh symmetry.

(3) Nodal point numbering.

Stress gradients should dictate the degree of fineness of the mesh. In areas of high stress gradients the mesh should be refined. Portions where stress levels are nearly constant can be represented with large elements in a coarse mesh. In many problems users intuitatively produce meshes that account for stress gradients. However, when complex shapes, boundary conditions and loads are involved, a preliminary coarse mesh analysis should be conducted. The indication of high stress gradients is not always available from the contour graphs, since they are based on averaged values of common nodal stresses. A true indication of the need for element refinement is when element stress levels at shared nodes differ considerably.

Many techniques for grading the mesh during initial construction, or when refining is required, are available. A comparison of two grading methods is shown in Figure 5.4.

Mesh symmetry with respect to some line or curve in the body should be maintained where possible. In complex shapes overall symmetry usually cannot be achieved. In these cases local symmetric areas should be constructed. Figure 5.2 illustrates symmetric mesh construction.

In a completed mesh the one element that has the largest difference between its nodal point numbers influences heavily the computation time required for the problem. The maximum difference in node numbers is an indication of the amount of storage required to house and solve the set of equilibrium equations. In solving the equations the data in storage is transferred, manipulated, and restored many times. The larger the data set, the longer the corresponding computation time. The mesh can be constructed so that it lends itself to optimum nodal point numbering technique, which in turn will reduce the largest nodal point number difference and the computation time.

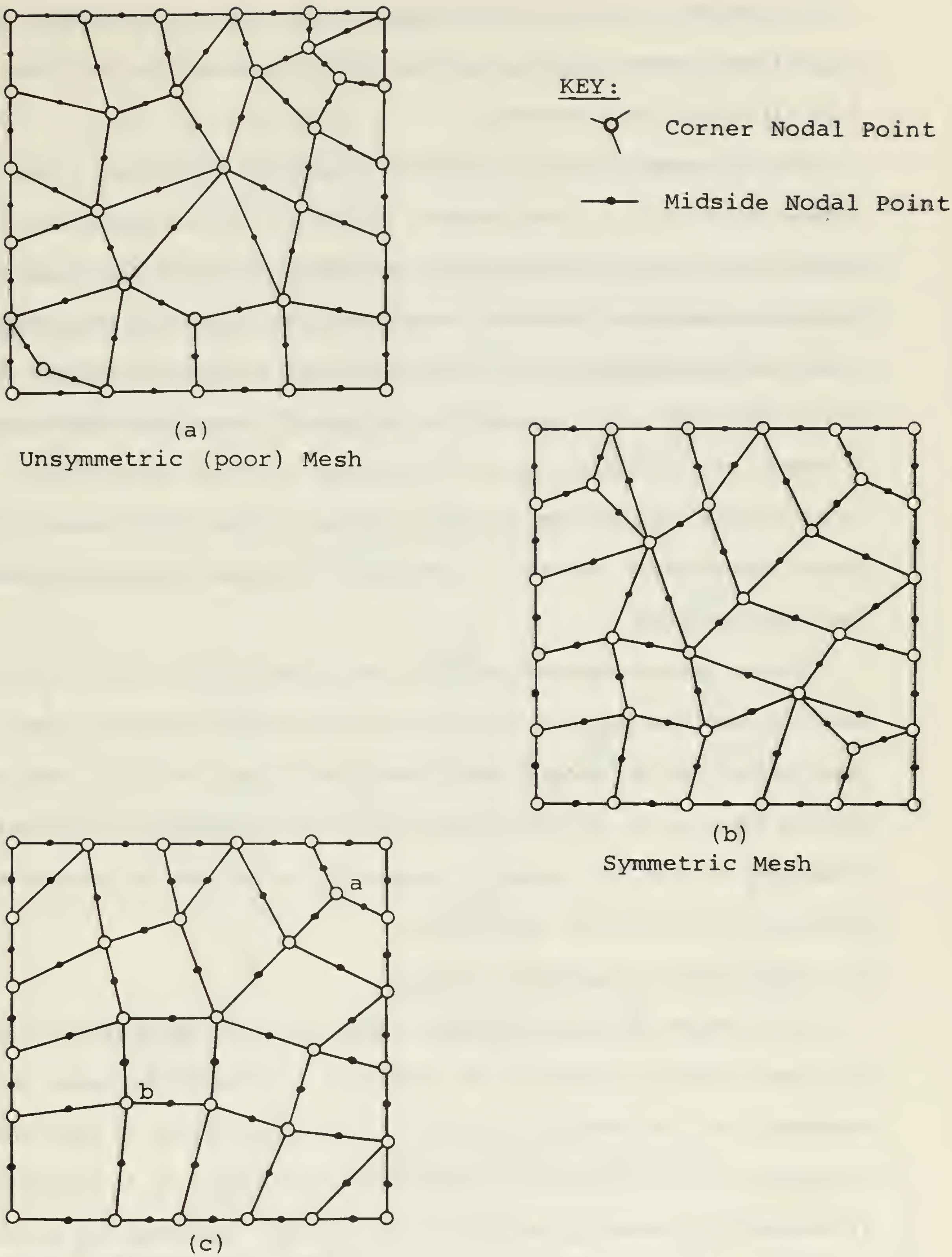


Figure 5.2 Symmetry in Mesh Construction

In general, the least nodal number difference can be achieved by placing all mesh corner nodes on lines or smooth curves within the body. Figure 5.2c illustrates this procedure.

It is suggested that the physical process of mesh construction be accompolished in the following manner. (1) Sketch initial mesh ideas on a scratch pad until the final version is formed. (2) Check the limitations on maximum numbers of elements, nodes, and constraints and the maximum nodal number difference. (3) Transfer the mesh layout to a large sheet where sufficient space is available to identify corner and midside nodes by number and to exhibit the X-Y coordinates of corner nodal points. Large (2'x3') Ozalid prints from a master drawing of faint ½ inch square grid with a superimposed quadrant of concentric arcs have been found convenient for this last step.

Meshes can be produced where all the elements are of equal size and have the same orientation. In this case all elements have the same stiffness matrix and the program need compute only one. This is a time saving feature that can be used for regular bodies or preliminary gross analyses. Frequently an auxiliary computer program can be written to generate all mesh input data cards in these cases.

5.5 Nodal point and element numbering

The program requires that each element and each nodal point be numbered for identification purposes. The numbering of elements and nodes is independent, but increasing both counts in the same pattern is convenient. Consecutive numbering is not a requirement, but again, it is convenient. If consecutive numbering is not used (occurs when combining two previously independent meshes) an ordered listing of the input sequence should be maintained, since output is identified by consecutive numbering.

The method of nodal point numbering is important for two reasons.

(1) To keep within the limit on maximum nodal point difference for a single element.

(2) Efficient numbering can significantly reduce computation time.

The general rule for efficient numbering is to number in a direction where the least number of nodes lie in "line" (not necessarily the smallest figure dimension) and repeat for the next "parallel line." The technique is illustrated in figures 5.3 and 5.4c.

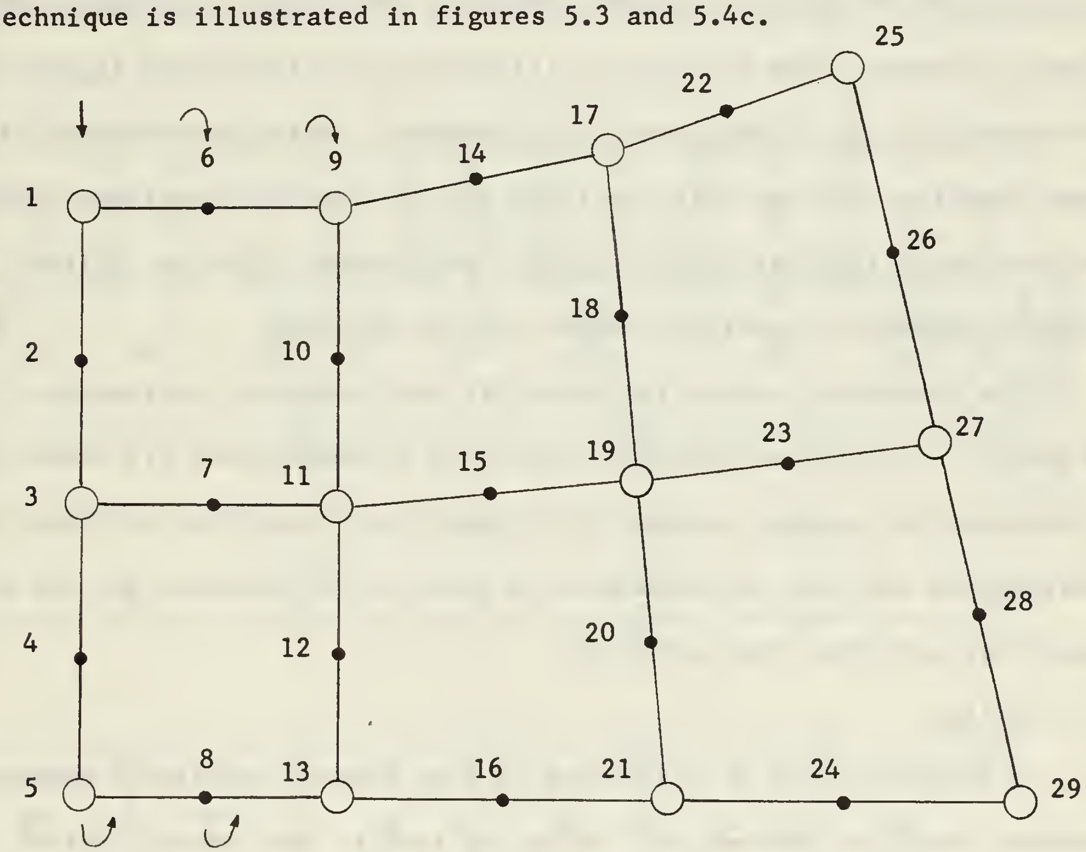


Figure 5.3 Nodal Numbering Technique

Efficient numbering within graded areas is often difficult, Figure 5.4 compares two grading techniques and three numbering methods. Figure 5.4(c) is preferred on both scores.

As discussed previously, the one element in the mesh with the greatest node number difference controls the solution time. If it is unavoidable that a large nodal number difference exists in some area of the mesh, nothing is to be gained by trying to reduce lower difference values elsewhere in the mesh.

5.6 Coordinates

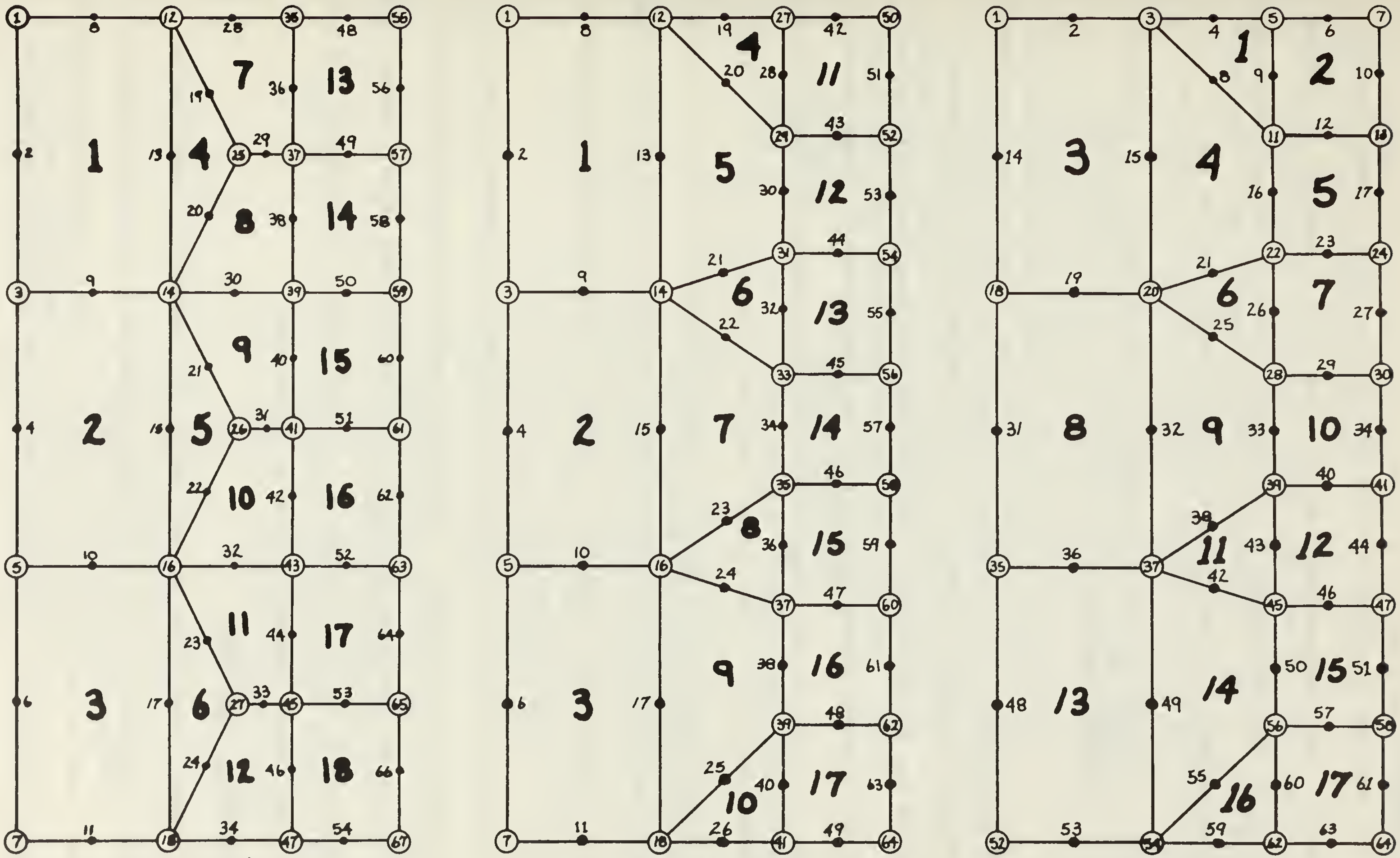
The global coordinate system for the program is the planar cartesian coordinate system. The X and Y coordinates of each corner point are required input data. This coordinate array data set is usually the largest single block of input data. Two hints are offered that can reduce the time to prepare these data cards. (1) Reduce the significant figures of the coordinate by slight nodal point movement. This procedure may disrupt mesh symmetry, but the effect will be small. (2) The coordinate origin can be placed internal to the figure. Mesh symmetry may be utilized to produce identical coordinate values, except for sign.

The coordinate system is compatible with structure overlapping. This situation has not been explored other than to demonstrate its dreadful influence on the contour graphs, which should not be printed in these cases. Overlapping can also be produced as a result of deformation and the program does not sense the condition.

5.7 Units

A consistent set of units must be used between the linear measure and surface traction load values. Where the inch is used as the unit of boundary and thickness measure, distributed forces and shears must be per square inch (eg. PSI). The modulus of elasticity and specific weight input constants must reflect the same units. The most useful set of units for mechanical design has been the inch and KSI (i.e. Kip per square inch); for large structural problems the foot and KSF.

(a) (b) (c)



Poor Grading
Poor Numbering

Good Grading
Poor Numbering

Good Grading
Good Numbering

Figure 5.4 Grading and Numbering Techniques

5.8 Materials

The program is written for homogenous, isotropic, linear elastic materials. The required input data are the Elastic Modulus, E; Poisson's Ratio, ν ; Specific Weight, γ ; and Coefficient of Thermal Expansion, α . For plane-strain problems reduced (primed) values must be used:

$$E' = \cancel{\frac{E}{(1-\nu)^2}}\ \frac{E}{(1-\nu^2)} \qquad \nu' = \frac{\nu}{1-\nu} \qquad \alpha' = \alpha(1+\nu)$$

The ability to utilize multiple materials is one of the advantages of the finite element procedure. In the program up to six different materials may be specified. True difference between materials may be used, or different property values of the same material. Temperature influence on the elastic properties may be approximated by using "different" materials when in the thermal gradient of a single material object. Body force loading, other than the program standard of 1-G, can be simulated by applying suitable multipliers to the value of the specific weight prior to input. This technique is useful in simulating dynamic inertia loading by a static equivalence.

When multi-material problems are analyzed, the element stress print option should be specified. The standard print of averaged nodal point stresses is not adequate at material interfaces. The stress contour graphs should be used only with judgement in these cases, since they do not reflect true stress discontinuities at interfaces.

5.9 Boundary conditions

Boundary conditions are applied at the appropriate nodal points (corner or midside) of the structure. A boundary constraint consists of fixing a node in one direction. A "fixed" boundary point will require two boundary constraints. The program will accept up to 250 constraints.

Constraints are not restricted to nodes on the physical boundary; however, this is the usual situation.

Initial values of boundary displacement may also be used as boundary conditions. This feature has many applications. One is the simulation of "poor fit" conditions in a part or structure. A second is illustrated in Figure 5.5, where actual gear tooth root area deflections were determined; then used as input initial displacements, thus freeing elements for use in a more refined mesh of the structure of interest.

In program input the boundary condition data is designated for the appropriate nodal point with a boundary condition "TAG." A list of TAG values and appropriate boundary conditions follow:

TAG VALUE	BOUNDARY CONDITION
TAG = 0	If the point is fixed in both directions. or Initial X and Y displacements are specified and the point is fixed in both directions.
TAG = 1	If the point is fixed in the X-direction and free to move in the Y-direction. or Initial X-displacement is specified but the point is free to move in the Y-direction.
TAG = 2	If the point is free to move along a line forming an angle ϕ with the positive X-axis. or Initial Y-displacement is specified but the point is free to move in the X-direction.

The boundary conditions are illustrated in Figure 5.6.

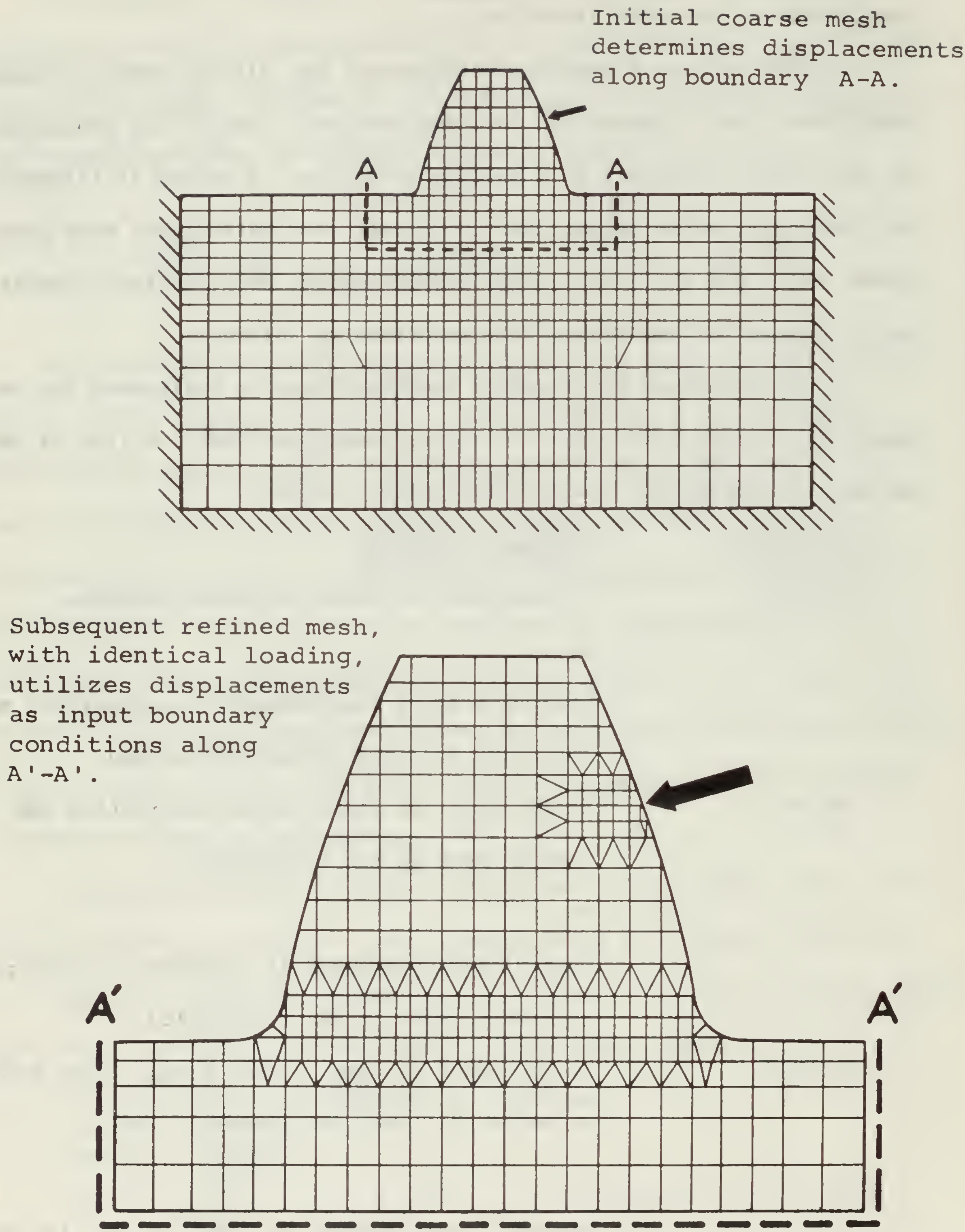


Figure 5.5 Use of Initial Displacement Boundary Conditions.

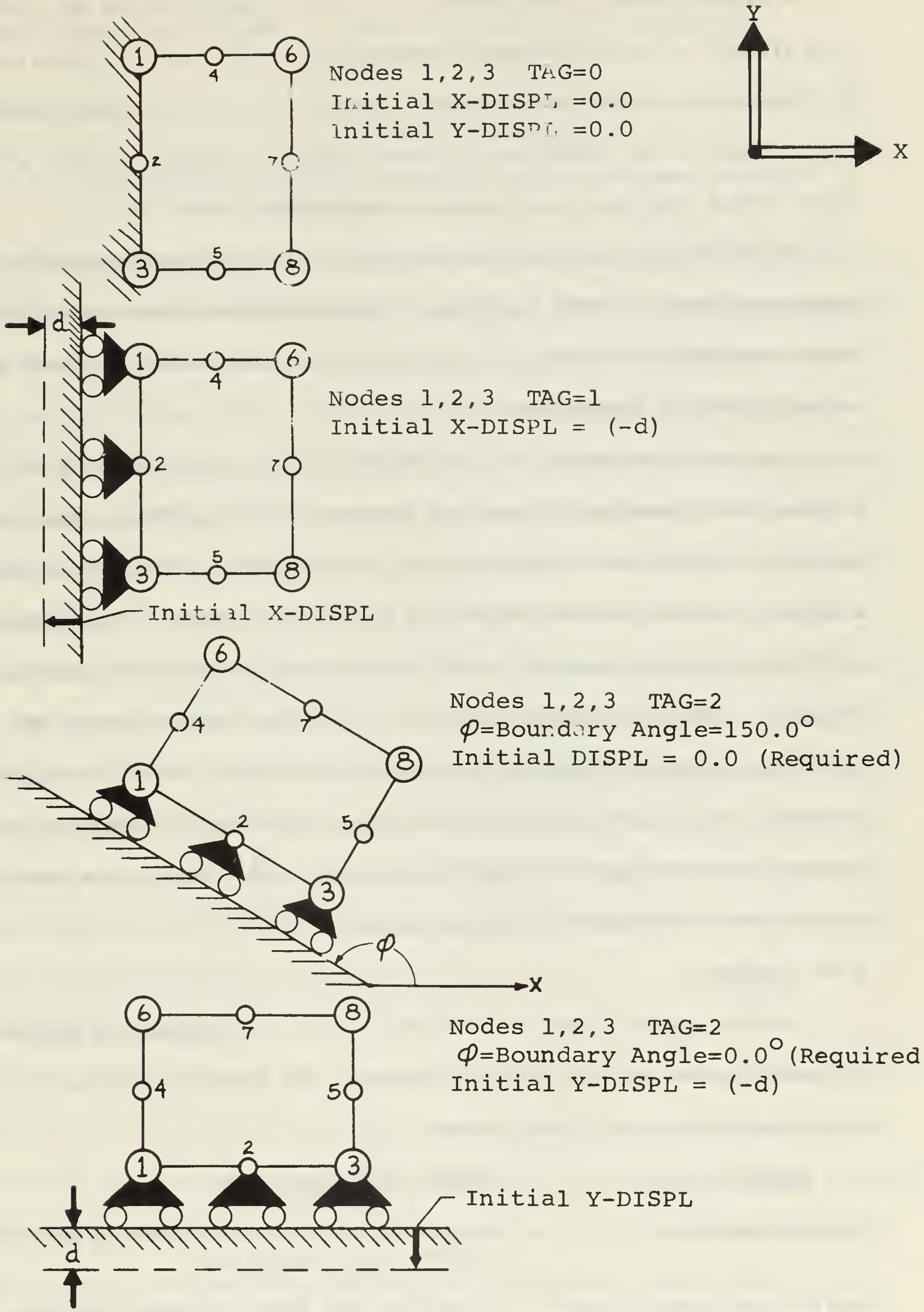


Figure 5.6 Boundary Conditions

A single initial displacement can only be specified in the X-direction (TAG=1) or the Y-direction (TAG=2, ϕ = 0.0°). Initial displacements for points on a slope roller are not available in this version of the program. There is an equivalance between TAG=1 and TAG=2, ϕ = 90.0°. The first method requires less internal computation.

In multi-load case problems any initial displacements specified will apply only for the first load case. Subsequent load cases will have the same constraint conditions for initially displaced nodes, but with a 0.0 value of initial displacement.

Boundary constraints are a fundamental characteristic of a structure and are known immediately upon the definition of a problem. When the advantages of 2-fold or 4-fold symmetry are utilized, with partial body analysis, the constraints needed are not always obvious. Care should be exercised in these cases to ensure duplication of the actual problem situation. When a complete structure is analyzed that is shape and load symmetric, auxiliary boundary constraints are often helpful to improve the solution. Frequently symmetric centerline nodes can be placed on rollers. This will avoid slight skewing displacements which result from round-off error in the stiffness matrix generation.

5.10 Loading

Three loading types are available with the program: (1) surface loads, (2) body forces, and (3) thermal loading. The general character of the input loading data is listed below:

Loading type	Input characteristics
Concentrated force	Magnitude of X and Y components specified. Loaded node identified.
Distributed normal force*	Magnitude and sense of normal pressure. Element side identified.
Distributed shear force*	Magnitude and sense of surface shear. Element side identified.

Gravity (body force) load	Indicated true (T) if applicable. Acts in (-Y) direction.
Thermal load	Indicate temperature increment at element corner nodes.

* - side variation assumed parabolic based on values specified at the nodes of the element side.

The sign convention for concentrated forces is that of the global coordinate system. Distributed forces follow a sign convention based on the element shape. Normal outward traction and counterclockwise shears are positive (Fig. 6.2). Gravity loading acts in the (-Y) direction.

Distributed forces and shears acting on an element side are assumed to have a parabolic variation based on the input values at the corner nodes and midside node involved. The resulting total element side load is represented by equivalent nodal point concentrated forces that are generated internally in the program. When the gravity load option is used, the program generates nodal forces equivalent to body forces to apply at all the mesh nodes. In thermal loading the program computes the forces that would be required to reestablish the size of the thermally expanded elements and applies them at the appropriate nodes. These computations are internal to the program, with no external indication other than the resulting displacements and stresses.

5.11 Contour graphs

The stress contour plots generated by the program are very useful. The plots are produced on a conventional printer, thus eliminating a requirement for specialized plotting equipment not available at many computer installations. The loss in definition due to printer plotting is compensated for by the short time required to produce the plots (about 9 sec.).

Six plots are available, each shows the body outline and contour lines of constant stress. The lines divide the full range of stresses encountered into ten equal increments. Plots of the following stresses are available; σ_x, σ_y, τ_{xy}, σ_{max}, σ_{min}, τ_{max}.

The plotting subroutine allows up to 50 elements to be eliminated from the area considered in determining the stress range for the contour lines. This feature allows (1) the plotting of only a portion of the structure, (2) elimination of high stress gradient areas that bunch the countour lines, (3) a combination of the two.

5.12 Error exits

Several error conditions caused by either wrong input data or exceeded program capacity are checked in subroutine SETUP. Error messages printed are self-explanatory and may be complemented by examination of the input data printout. The program does not stop until all error conditions have been tested. If another problem follows after an error detection, the program searches for the next START indicator card at which time execution continues on the new problem.

5.13 Displacement solution iterations

Double precision iterations for improvement of the displacement solution are optionally available in the program. One or two iterations should always be used for large ill-conditioned problems. Each iteration is essentially re-solving the displacement equations to eliminate round-off error, and will take approximately 90% of the original problem solution time.

5.14 Card punching of displacements and stresses

A program option is the punching of nodal displacements and averaged nodal point stresses. These card data sets may be used as input to plotting equipment for the production of more sophisticated plots than produced by the program.

5.15 Timing

Sufficient problems have not been run with the program to develop an accurate empirical relationship giving overall execution time. An estimator currently in use is:

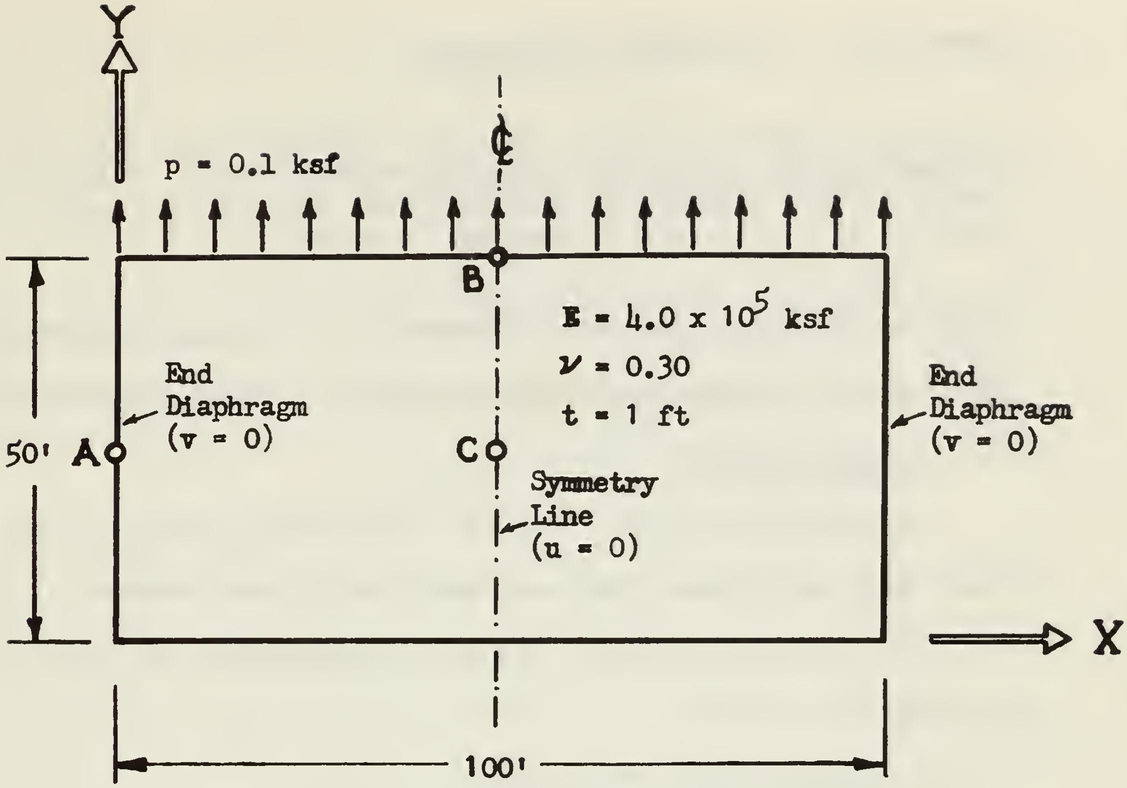
	60 nodes, half-bandwidth < 60
1 minute execution time for each: (no iterations)	50 nodes, $60 \leqslant$ half-bandwidth < 120
	40 nodes, $120 \leqslant$ half-bandwidth

For small problems (number of elements < 40) execution times predicted will be low since the subroutine SETUP requires a minimum of 50 sec. regardless of problem size. Timing estimates apply to an IBM OS/360 computer.

5.16 Accuracy

To illustrate the accuracy of this version, a 2:1 symmetric plate under uniform in-plane load on top (Fig. 5.7) and constrained by diaphragms at the ends (X-disp. only) was analyzed by subdividing one-half of the plate into meshes of (n x n) quadrilaterals - square in this case. There is no known analytic solution, but this problem has been used extensively in comparison tests of different types of finite elements because of the diversity of stress conditions and the ease of preparation. Precision is not sought in the analysis, since a "true" solution is guaranteed if the mesh is sufficiently refined. Rapidity of convergence is the desirable characteristic.

A comparison of the typical values reproduced in Figure 5.7 shows that the 8 x 8 mesh provides almost 5 decimal digits for the displacements, 4 for σ_x and 3 for τ_{xy}. Actually, the program has capacity for an 18 x 18 mesh (5500 degrees of freedom) if necessary. The consistency of the stress values is reflected by the fact that the maximum discrepancy over contributing elements nodal points did not exceed 0.004 for σ_x and σ_y, and 0.006 for τ_{xy} in the case of the 8 x 8 mesh.



Mesh of square elements for one-half of plate	2 x 2	4 x 4	8 x 8	Asymptotic Extrapolated Values
Degrees of Freedom:				
unconstrained	42	290	1090	-
after B.C.s	32	272	1056	-
Deflection $v_C \times 10^5$	4.8297	4.8487	4.8515	4.8516
Normal Stress σ_{xB}	0.3190	0.3207	0.3214	0.3215
Shear Stress τ_{xyA}	0.1550	0.1380	0.1333	0.1321

Fig. 5.7 - Example: 2:1 Symmetric Plate

Chapter VI Input Data Preparation

This chapter is written as a self contained unit giving all specific information for the preparation of input data cards. When facility with the use of the program has been attained, this chapter may be duplicated and used as a condensed program user's manual.

FLOATING POINT (F) FORMATS ARE COMPATIBLE WITH ASSIGNED EXPONENT (E) FORMATS

RIGHT JUSTIFY INTEGER AND EXPONENT NUMBERS IN THEIR ASSIGNED FIELDS

6.1 Structure data

(a) Start Card (A8): With the word START punched in cols. 1-5. This card must precede the input data deck of any problem.

(b) Title Card (20A4): Alphameric information in cols. 1-80 to identify the output.

(c) Control Card (8I4, 6L2):

Columns	Variable Name	Meaning
1-4	NUMEL	Number of elements ($\leq$ 350);
5-8	NUMCP	Number of corner points;
9-12	NUMNP	Number of total nodal points ($\leq$ 1050);
13-16	NUMBC	Number of restrained nodal points;
17-20	NUMPB	Number of plot defining boundary points, see item (e);
21-24	NLOAD	Number of load cases; will be set=1 if left blank;
25-28	NMAT	Number of different materials ($\leq$ 6); will be set=1 if left blank;
29-32	MAXIT	Maximum number of residual iterations in the displacement solution; punch a 1 or 2 for large, ill-conditioned problems.

The next five fields are for logical flags; if a T is punched, in any assigned column, the indicated option takes place. A blank of F implies FALSE.

33-34	T1	All quadrilaterals have the same stiffness matrix (see note 1);
35-36	T2	Punching of mesh nodal point coordinates and displacements (I4, 2F8.3, 2E14.5);
37-38	T3	Punching of averaged σ_x, σ_y, and τ_{xy} at mesh nodal points and quadrilateral centroids (I4, 3E18.6);
39-40	T4	Print of element stresses (see note 2);
41-42	T5	Another complete problem follows.

<u>Notes</u>

(1) All quadrilaterals have the same stiffness if they can be superimposed by a translation.

(2) Element stresses should be printed in problems involving several material types, since averaged stresses and their plots do not display actual interface discontinuities.

(d) <u>Material Property Table</u> (I4, E10.3, 2F10.3, E10.4): One card per material type (total NMAT cards):

Cols.	1-4	Material type number;
	5-14	Elastic modulus;
	15-24	Poisson's ratio;
	25-34	Specific weight;
	35-44	Coefficient of thermal expansion.

For plane-strain, reduced values must be used;

$$E' = \frac{E}{(1-\nu^2)} \qquad \nu' = \frac{\nu}{(1-\nu)} \qquad \alpha' = \alpha(1+\nu)$$

(e) <u>Plotted figure boundary outline array</u> (20I4): For the plotting of stress graphs, NUMPB number of <u>corner</u> nodal points which, when connected outline the structure, or structure subregion, as a series of straight lines must be punched, in cyclic order, 20 node numbers per card. The starting corner node and direction of travel around the structure, or subregion, boundary are arbitrary. Holes in multiply connected bodies cannot be outlined separately. In subregion plots, the elements ($\leqslant$ 50) outside the plotted figure boundary (skipped elements) must be listed in item (1) under Loading.

(f) Element nodal point array (10I4, F10.3): One card per element. (total NUMEL cards).

Cols. 1-4 Element number;

5-36 Nodal point numbers:

(I) for quadrilateral: element nodal points in counterclockwise order I-J-K-L-M-O-P (Fig. 6.1a). The starting corner is arbitrary, except when equal stiffnesses are implied (i.e., T1=T in Control Card). In this case the starting corners must be in the same location for all elements.

(II) for single triangles: punch nodes I-J-K-L-M-N, (Fig. 6.1b), leave cols. 29-36 blank.

37-40 Element material type, will be set=1 if left blank.

41-50 Element thickness, will be set=1.0 if left blank.

Note: If a quadrilateral is not convex (not recommended) the entrant corner must be either J or K.

(g) Corner point coordinate array (I4, 2F8.3): One card per corner point (total NUMCP cards).

Cols. 1-4 Corner nodal point number;

5-12 X-coordinate

13-20 Y-coordinate

(h) Boundary condition array (2I4, 2E15.3): One card per restrained nodal point (total NUMBC cards).

Cols. 1-4 Nodal point number

5-8 Tag = 0 if point fixed in both directions or initial displacement is specified in both the X and Y direction then point is fixed.

= 1 if point is fixed in the X-direction and free in the Y-direction or initial displacement is specified in the X-direction and point is free in the Y-direction.

= 2 if point is free to move along a line forming angle ϕ with the X-axis and fixed in a direction normal to that line; or if initial displacement is specified in the Y-direction and the point is free in the X-direction.

9-23 Angle in degrees, positive counterclockwise from X-axis for type TAG=2 boundary condition.

Initial X-displacement value for TAG=0 boundary condition (for both cases will be set=0.0 if left blank)

24-38 Initial displacement boundary condition value. Y-displacement for TAG=0 or 2. X-displacement for TAG=1. (will be set = 0.0 if left blank)

6.2 Loading data

Each load case must be specified by a data deck initiated by a LOADING card; this package follows the structure data deck. A load deck consists of the following cards;

(i) Loading Card (A8): With the word LOADING punched in Cols. 1-7.

(j) Title Card (20A4): Alphameric information in cols. 1-80 to identify the load case.

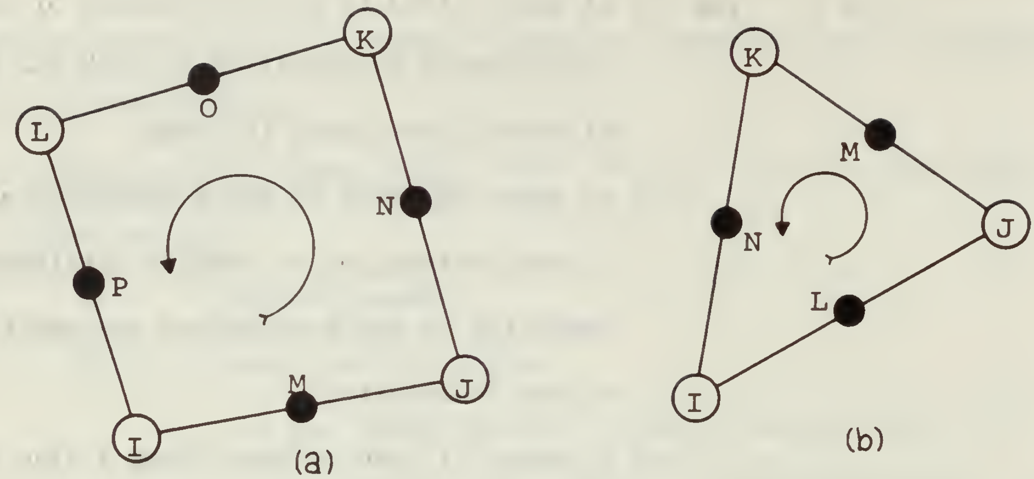
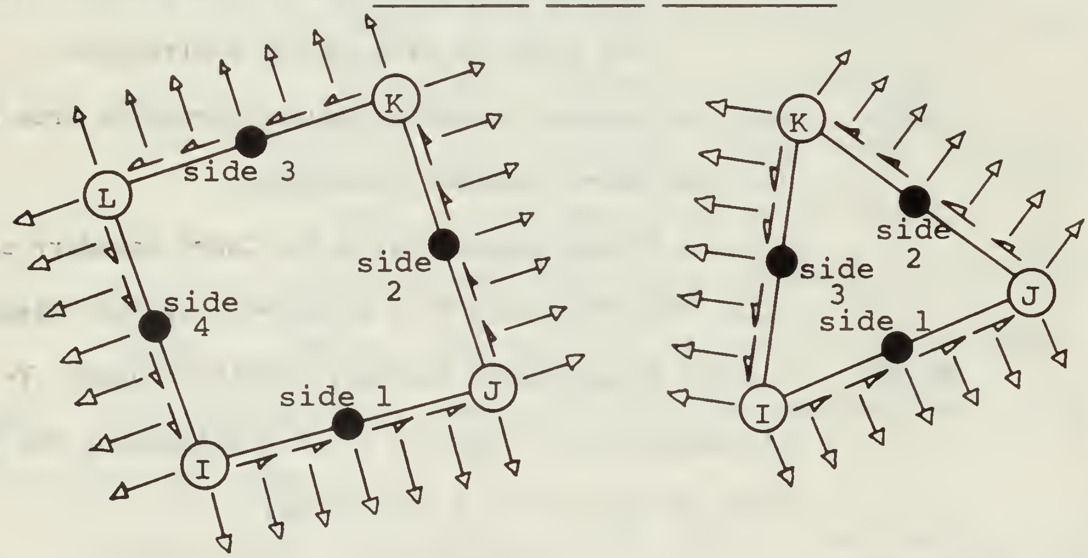
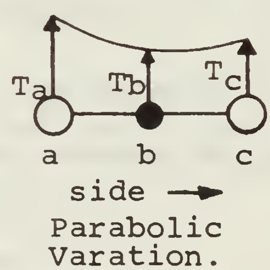


Figure 6.1 Element Nodal Point Identification

Positive Sense Indicated





Quadrilateral

	a	b	c
side 1	I	M	J
side 2	J	N	K
side 3	K	O	L
side 4	L	P	I

Triangle

	a	b	c
side 1	I	L	J
side 2	J	M	K
side 3	K	N	I

Figure 6.2 Convention for Element Side Loading

(k) <u>Control card</u> (3I4,L2)

Columns	Variable Name	Meaning
1-4	NPLD	Number of nodal points with concentrated forces;
5-8	NELD	Number of element sides loaded with distributed forces;
9-12	NTLD	Number of elements undergoing thermal increments;
13-14	DENS	Logical flag for gravity loading: If a T (implies .TRUE.) is punched, gravity forces acting along the (-Y) direction are considered.

(1) <u>Stress Contour Graph Indicator Card</u> (7I4): A <u>positive</u> <u>integer</u> punched in any of the first six fields will cause a stress contour plot to be produced, on the printer, for the indicated stress component.

Columns	Graph
1-4	Sigma x
5-8	Sigma y
9-12	Tau xy
13-16	Sigma max
17-20	Sigma min
21-24	Max shear

The last field (cols. 25-28) indicates the number NSK $\leqslant$ 50 of elements to be skipped from the plots. If NSK > 0, additional cards must follow, specifying the skipped element numbers (20I4). Element skipping may be used for two different purposes:

(1) to eliminate small regions of high stress gradients which cannot be accurately described by a printer plot;

(2) to plot a portion of the structure, which is then amplified. In this case the plotted figure outline array (item e) must specify the outline of the subregion.

(m) Nodal Point Forces (I4, 2F8.3): One card per nodal point loaded with a concentrated force (no cards if NPLD = 0).

Cols.	1-4	Nodal point number;
	5-12	X-load;
	13-20	Y-load.

(n) Element Side Loads (2I4, 6F8.3): One card per element side under surface traction (no cards if NELD = 0). The convention for positive traction and shear is indicated in Figure 6.2. The side variation is assumed to be parabolic and specified by the values at points a, b and c (in counterclockwise sense). For instance, for side 2 of a quadrilateral: a = corner point J, b = midside point N, c = corner point K.

Cols.	1-4	Element number;
	5-8	Side number (see Fig. 6.2);
	9-16	Normal traction at a (T_a);
	17-24	Normal traction at b (T_b);
	25-32	Normal traction at c (T_c);
	33-40	Surface shear at a;
	41-48	Surface shear at b;
	49-56	Surface shear at c.

These values must be specified per unit of length of the figure and per unit of thickness, (eg., PSI).

(o) Thermal Increments (I4, 4F10.3): One card per element undergoing temperature changes (no cards if NTLD = 0):

Cols.	1-4	Element number;
	5-14	Temperature variation at corner I;
	15-24	Temperature variation at corner J;
	25-34	Temperature variation at corner K;
	35-44	Temperature variation at corner L (leave blank for a triangle).

Note: the thermal increment at the centroid of quadrilaterals is assumed to be the mean of the corner values, and a linear variation assumed over each subtriangle.

6.3 New problem

The input of a new problem must follow the last load deck for the previous one. For safety, any number of blank cards may be inserted before the START card.

Recommendations

The most hearty recommendation is to use the program. It represents a powerful engineering tool.

Recommendations for augmentation of the method are:

(1) Inclusion of an optional bilinear element stiffness subroutine to allow analysis of the ever increasing group of materials that have such characteristics.

(2) Modification to accept linearly varying element thickness.

(3) A general orthotropic element stiffness subroutine.

Recommendations for augmentation of the program are:

(1) A mesh generation package configured for the NPS Computer Center IBM Optical Display Unit.

(2) A contour graph plotting package for use on commercial X-Y plotters, and compatible with the program punched card output of displacements and stresses.

(3) Conversion of tape-disk external storage statements in the program to direct access statements.

References

1. Cantin, G., "A Curved Finite Element for Cylindrical Shells," Ph.D. Dissertation, University of California, Berkeley, 1967
2. Clough, R. W., "The Finite Element Method in Structural Mechanics," Chapter 7 of "Stress Analysis," edited by O. C. Zienkiewicz and G. S. Holister, John Wiley and Sons, 1965
3. Felippa, C. A., "Refined Finite Element Analysis of Linear and Non linear Two-dimensional Structures," SEL Report No. 66-22, University of California, Berkeley, 1966
4. Fox, L., "An Introduction of Numerical Linear Algebra," Oxford University Press, 1965
5. Fraeijis de Veubeke, B. M., "Displacement and Equilibrium Models in the Finite Element Method," Chapter 9 of "Stress Analysis," cited in reference 2.
6. IBM System/360, "FORTRAN IV Language," Form C-28-6515-5, IBM Corp., Poughkeepsie, N. Y.
7. IBM System/360 Operating System, "FORTRAN IV (H) Programmer's Guide," Form C-28-6602-1, IBM Corp., Poughkeepsie, N. Y.
8. IBM System/360 Operation System, "Linkage Editor," Form C-28-6538-4, IBM Corp., Poughkeepsie, N. Y.
9. Johnson, C. P., "The Analysis of Thin Shells by a Finite Element Procedure," SEL Report No. 67-22, University of California, Berkeley, 1967
10. Mc Cormic, C. W., Herbert, K. J., "Solution of Linear Equations with Digital Computers," California Institute of Technology, Pasadena, 1965

11. Melosh, R. J., "Development of the Stiffness Method to Define Bounds on Elastic Behavior of Structures," Ph.D. Dissertation University of Washington, 1962

12. Turner, M. J., "The Direct Stiffness Method of Structural Analysis," AGARD Meeting, Aachen, Germany, 1959

13. Turner, M. J., Clough, R. W., Martin, H. C., and Topp, L. J., "Stiffness and Deflection Analysis of Complex Structures," J. Aeron Sci., Vol. 23, No. 9, 1956

14. Turner, M. J., Martin, H. C. and Weikel, R. C., "Further Developments and Amplifications of the Stiffness Method," AGARDograph NO. 72, edited by F. de Veubeke, Pergamon Press, 1964

15. University of California, "Solution of Linear Equations," Div. of Structural Engineering and Structural Mechanics, Report No. CE 290G

16. Zienkiewicz, O. C., "The Finite Element in Structural and Continuum Mechanics," McGraw-Hill Publishing Co. Ltd., London, 1967

APPENDIX 1 Sample Problem - "Flounder" plate in tension

This appendix presents a sample problem analyzed with the program. The structure and loading situation are somewhat hypothetical to allow presentation of a variety of input data and computed results. Actual input data decks and computer output are illustrated.

A "Flounder" plate in the Naval service is any roughly triangular plate, that in its lifetime may inadvertently find itself on the ocean floor. The example problem presented in this appendix is the analysis of one type of flounder plate.

The function of the problem plate is that of an attachment member. The plate is welded to a "Trolley-Block" apparatus which rides, on cables, between ships at sea conducting underway transfer of supplies. The flounder plate acts as the attachment point for a small fixture, which in turn holds nets or boxes containing the transferred material. The problem is the preliminary coarse mesh analysis of the plate.

Sketches of the complete trolley-block assembly and the loaded flounder plate are presented in Figure A1.1. The figure also illustrates the idealized form of the plate; nodal point numbers; element numbers; nodal coordinates; boundary conditions, and input loading.

The program input data deck is illustrated in Figure A1.12. Data values that are overlined in the figure represent input for which optional default values are available. The size of the data fields for input data are also illustrated.

The computed output data is presented in Figure A1.3, where significant items are annotated on the figure.

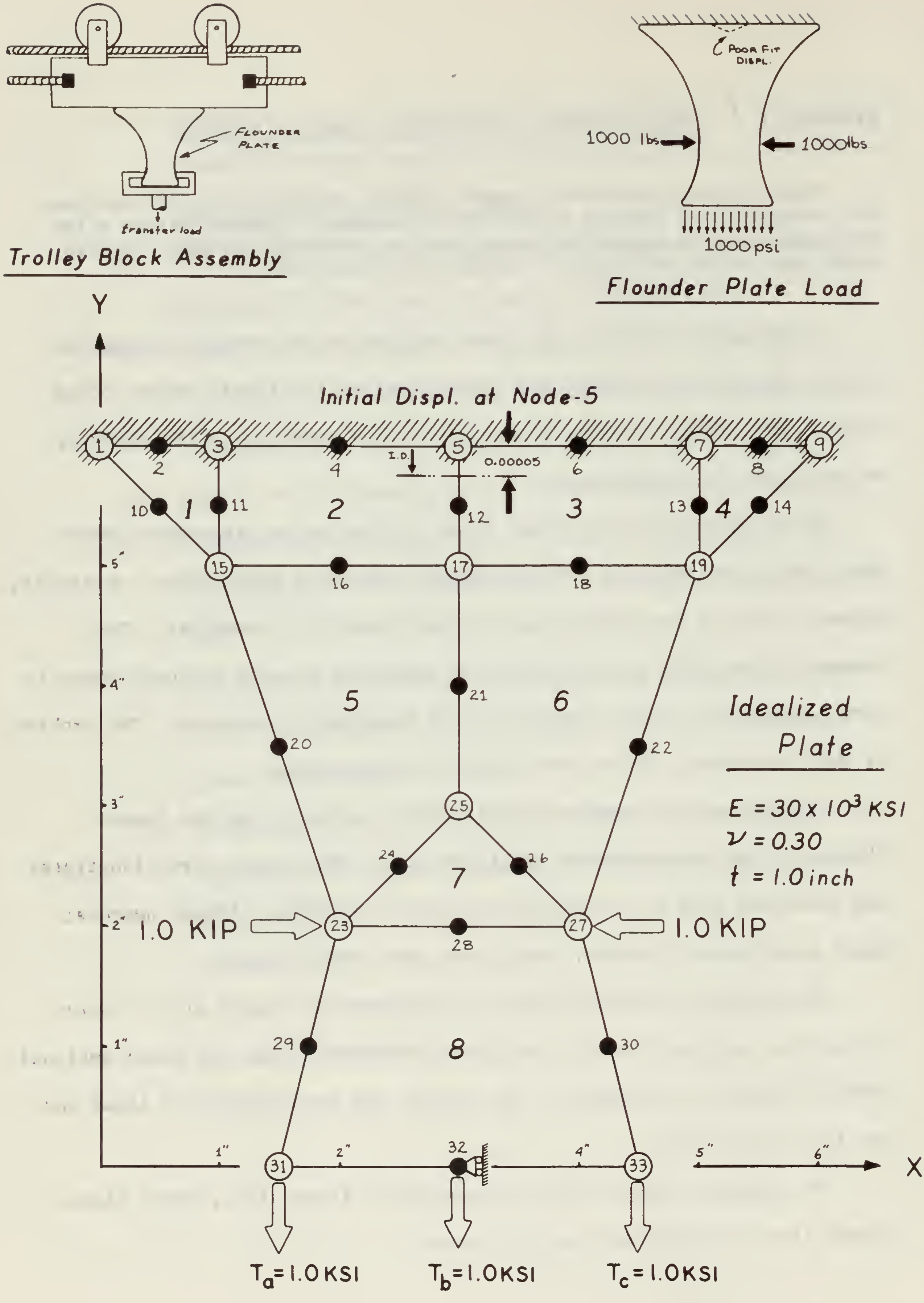


FIGURE A1.1 SAMPLE PROBLEM STRUCTURE

```
START
MALONE BOX-30        EXAMPLE PROBLEM - FLOUNDER PLATE        UNITS: INCH/KIP/KSI
  8  13  33  10  11   1   1   0 F F F T F
  1   30.0E+03    0.30    0.286   7.0E-06
  1   3   5   7   9  19  27  33  31  23  15
  1  15   3   1  11   2  10              1        1.0
  2  15  17   5   3  16  12   4  11
  3  17  19   7   5  18  13   6  12
  4  19   9   7  14   8  13
  5  23  25  17  15  24  21  16  20
  6  25  27  19  17  26  22  18  21
  7  23  27  25  28  26  24
  8  31  33  27  23  32  30  28  29
  1     0.00    6.00
  3     1.00    6.00
  5     3.00    6.00
  7     5.00    6.00
  9     6.00    6.00
 15     1.00    5.00
 17     3.00    5.00
 19     5.00    5.00
 23     2.00    2.00
 25     3.00    3.00
 27     4.00    3.00
 31     1.50    0.00
 33     4.50    0.00
  1   0            0.0          0.0
  2   0
  3   0
  4   0
  5   2            0.0       -0.00005
  6   0
  7   0
  8   0
  9   0
 32   1
LOADING
MALONE BOX-30        EXAMPLE PROBLEM - FLOUNDER PLATE        UNITS: INCH/KIP/KSI
  2   1   0 F
  1   1   1   1   1   1   0
 29     1.0     0.0
 30    -1.0     0.0
  8   1   +1.0   +1.0   +1.0   0.0   0.0   0.0
```

FIGURE A1.2   SAMPLE PROBLEM COMPUTER INPUT DECK

```
MALONE BOX-30      EXAMPLE PROBLEM - FLOUNDER PLATE       UNITS: INCH/KIP/KSI

NUMBER OF ELEMENTS . . . . . . . . .          8
NUMBER OF CORNER POINTS . . . . . .          13
NUMBER OF NODAL POINTS  . . . . . .          33
NUMBER OF BOUNDARY CONDITIONS . . .          10
NUMBER OF DEF. BOUNDARY POINTS  . .          11
NUMBER OF LOAD CASES. . . . . . . .           1
NUMBER OF DIFFERENT MATERIALS . . .           1
MAX NO OF RESIDUAL ITERATIONS . . .           0

FLAG EQUAL TYPE QUADRILATERALS. . .           F
FLAG PUNCH DISPLACEMENTS. . . . . .           F
FLAG PUNCH STRESSES . . . . . . . .           F
FLAG PRINT ELEMENT STRESSES . . . .           T
FLAG NEW JOB FOLLOWS  . . . . . . .           F

MATERIAL PROPERTIES

MAT. NO.     ELASTIC MODULUS     POISSON'S RATIO        DENSITY      THERMAL EXPANSION COEFF.

   1         30.00000E 03           0.30000             0.28600            7.00000E-06

DEFINING BOUNDARY POINTS

  1    3    5    7    9   19   27   33   31   23   15
```

Note: The total execution time for this problem was 1.427 min. The 66 equations were solved in 8.895 sec.

FIGURE A1.3 SAMPLE PROBLEM COMPUTER OUTPUT

ELEMENT ARRAY

ELEMENT		I	J	K	L	M	N	O	P	MAT.TYPE	THICKNESS
TRIANGLE	1	15	3	1	11	2	10			1	1.0000
QUAD'LTERAL	2	15	17	5	3	16	12	4	11	1	1.0000
QUAD'LTERAL	3	17	19	7	5	18	13	6	12	1	1.0000
TRIANGLE	4	19	9	7	14	8	13			1	1.0000
QUAD'LTERAL	5	23	25	17	15	24	21	16	20	1	1.0000
QUAD'LTERAL	6	25	27	19	17	26	22	18	21	1	1.0000
TRIANGLE	7	23	27	25	28	26	24			1	1.0000
QUAD'LTERAL	8	31	33	27	23	32	30	28	29	1	1.0000

CORNER POINT COORDINATES

POINT	X-ORD	Y-ORD	POINT	X-ORD	Y-ORD
1	0.0	6.0000	3	1.0000	6.0000
5	3.0000	6.0000	7	5.0000	6.0000
9	6.0000	6.0000	15	1.0000	5.0000
17	3.0000	5.0000	19	5.0000	5.0000
23	2.0000	2.0000	25	3.0000	3.0000
27	4.0000	2.0000	31	1.5000	0.0
33	4.5000	0.0			

BOUNDARY CONDITIONS

NODAL POINT	TAG VALUE	BOUNDARY ANGLE (TAG=0, X-DISP)	INIT. BOUNDARY DISPLACEMENT
1	0	0.0	0.0
2	0	0.0	0.0
3	0	0.0	0.0
4	0	0.0	0.0
5	2	0.0	-0.0000500000
6	0	0.0	0.0
7	0	0.0	0.0
8	0	0.0	0.0
9	0	0.0	0.0
32	1	0.0	0.0

```
LOAD CASE NO.     1

 MALONE BOX-3C      EXAMFLE PROBLEM - FLOUNDER PLATE       UNITS: INCH/KIP/KSI

NO. CF NODAL POINT LOAD CARDS . .          2
NO. OF ELEMENT LOAD CARDS . . . .          1
NO. OF THERMALLY LOADED ELEMENTS           0
FLAG FOR GRAVITY LCADING  . . . .          F

 GRAPH               TAG
            (1=PLOT,C=NC PLCT)
 SIGMA XX            1
 SIGMA YY            1
 TAU XY              1
 SIGMA MAX           1
 SIGMA MIN           1
 MAX SHEAR           1

CONCENTRATED NODAL FORCES

PCINT         X-LOAD        Y-LCAD

  29          1.C0000        0.C
  30         -1.C000C        C.0

ELEMENT SICE FORCES

ELEMENT    SIDE    NODE   N. PRESSURE   SURF. SHEAR

     8       1      31        1.CCOO        0.0
                    32        1.CCOC        0.0
                    33        1.CCOO        C.0
```

NODAL FORCE VECTOR

POINT	X-LOAD	Y-LOAD	POINT	X-LOAD	Y-LOAD
1	0.0	0.0	2	0.0	0.0
3	0.0	0.0	4	0.0	0.0
5	1.22070E-07	-5.00000E-05	6	0.0	0.0
7	0.0	0.0	8	0.0	0.0
9	0.0	0.0	10	0.0	0.0
11	5.36474E-04	3.84940E-02	12	-1.95312E-07	-2.08123E 00
13	-5.36462E-04	3.84939E-02	14	0.0	0.0
15	1.18337E-02	4.88728E-02	16	-1.21353E-01	-2.10914E-01
17	-1.70898E-07	2.27305E-01	18	1.21353E-01	-2.10914E-01
19	-1.18338E-02	4.88728E-02	20	0.0	0.0
21	0.0	0.0	22	0.0	0.0
23	0.0	0.0	24	0.0	0.0
25	0.0	0.0	26	0.0	0.0
27	0.0	0.0	28	0.0	0.0
29	1.00000E 00	0.0	30	-1.00000E 00	0.0
31	0.0	-5.00000E-01	32	0.0	-2.00000E 00
33	0.0	-5.00000E-01			

Note: Initial Displacement values appear in the Force Vector at their Node and direction of application. See Y-LOAD column, at Node-5.

NCCAL PCINT CISPLACEMENTS

PCINT	X-DIS	Y-DIS	PCINT	X-DIS	Y-DIS
1	0.0	0.0	2	0.0	0.0
3	0.0	0.0	4	0.0	0.0
5	-9.43601E-12	-5.00000E-05	6	0.0	0.0
7	0.0	0.0	8	0.0	0.0
9	0.0	0.0	10	5.16003E-06	-2.53046E-06
11	6.18801E-06	-6.16182E-06	12	-1.75078E-11	-4.11088E-05
13	-6.18801E-06	-6.16184E-06	14	-5.16003E-06	-2.53047E-06
15	9.10618E-06	-1.30548E-05	16	3.28104E-06	-3.06653E-05
17	-4.55316E-11	-4.36710E-05	18	-3.28106E-06	-3.06652E-05
19	-9.10620E-06	-1.30549E-05	20	1.00769E-05	-6.31234E-05
21	-6.27551E-11	-6.71460E-05	22	-1.00770E-05	-6.31233E-05
23	1.04053E-05	-1.39917E-04	24	3.36744E-06	-1.17582E-04
25	-1.34946E-10	-9.92751E-05	26	-3.36759E-06	-1.17582E-04
27	-1.04056E-05	-1.39917E-04	28	-3.36513E-11	-1.36594E-04
29	4.17094E-05	-1.97021E-04	30	-4.17096E-05	-1.97021E-04
31	3.51312E-05	-2.37099E-04	32	0.0	-2.28481E-04
33	-3.51315E-05	-2.37099E-04			

<u>Note:</u> The slight skewing X-direction displacements at Nodes 5, 12, 17, 21, 25, and 28. This characteristic (due to round-off) was prevented at Node 32 by the application of an auxiliary boundary condition.

ELEMENT STRESSES

ELEMENT		N' POINT		SIG-XX	SIG-YY	TAU-XY
TRIANGLE 1		CORNER	15	0.27913	0.51925	-0.19721
		CORNER	3	0.11465	0.38217	-0.18053
		CORNER	1	-0.02901	-0.09669	-0.13308
QUADRILATERAL	2	CORNER	15	0.0473	0.6641	-0.1712
		CORNER	17	0.1844	0.5001	-0.1362
		CORNER	5	0.1579	-0.1807	-0.5774
		CORNER	3	0.2900	0.7573	-0.0024
		CENTROID		0.0210	1.1787	-0.2954
QUADRILATERAL	3	CORNER	17	0.1844	0.5001	0.1862
		CORNER	19	0.0473	0.6641	0.1712
		CORNER	7	0.2900	0.7573	0.0024
		CORNER	5	0.1579	-0.1807	0.5774
		CENTROID		0.0210	1.1787	0.2954
TRIANGLE 4		CORNER	19	0.27913	0.51926	0.19722
		CORNER	9	-0.02901	-0.09669	0.13308
		CORNER	7	0.11465	0.38217	0.18053
QUADRILATERAL	5	CORNER	23	0.0856	1.6063	-0.1834
		CORNER	25	0.1624	1.0867	0.0288
		CORNER	17	0.1365	0.6785	-0.0914
		CORNER	15	0.0484	0.5944	-0.3294
		CENTROID		0.0849	0.9752	-0.1852
QUADRILATERAL	6	CORNER	25	0.1624	1.0867	-0.0288
		CORNER	27	0.0856	1.6063	0.1834
		CORNER	19	0.0484	0.5944	0.3294
		CORNER	17	0.1365	0.6785	0.0914
		CENTROID		0.0849	0.9751	0.1852
TRIANGLE 7		CORNER	23	0.07285	1.28336	-0.00801
		CORNER	27	0.07284	1.28338	0.00802
		CORNER	25	0.22125	1.04397	0.00000
QUADRILATERAL	8	CORNER	31	-0.5696	0.7756	0.4820
		CORNER	33	-0.5696	0.7757	-0.4820
		CORNER	27	-0.1789	1.4426	0.1944
		CORNER	23	-0.1789	1.4426	-0.1944
		CENTROID		-0.6816	1.4006	0.0000

Note: The need for mesh refinement at Nodes where element stress values differ considerably. Example: Node-5 in elements 2 and 3, where a complete reversel of shear is indicated.

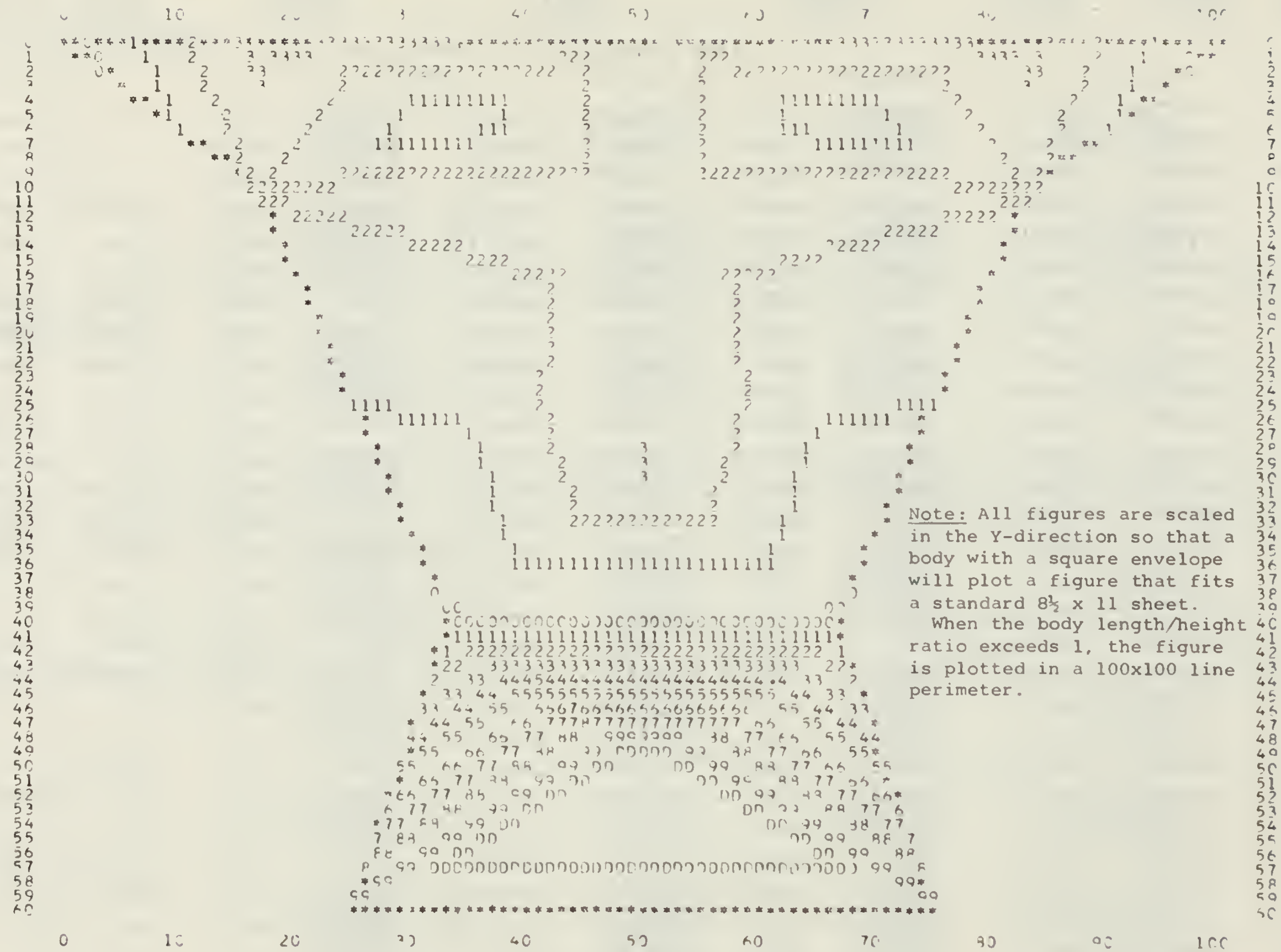
AVERAGED NODAL POINT STRESSES

POINT	COORDINATES X	COORDINATES Y	SIGMA-XX AVERAGE	SIGMA-XX MAXIMUM	SIGMA-YY AVERAGE	SIGMA-YY MAXIMUM	TAU-XY AVERAGE	TAU-XY MAXIMUM	SIG-MAX	SIG-MIN	MAX. SHEAR	ANGLE (SIG-MAX,X)
1	0.0	6.000	-0.0290	-0.0290	-0.0967	-0.0967	-0.1331	-0.1331	0.0745	-0.2002	0.1373	-37.8663
2	0.500	6.000	0.0428	0.0428	0.1427	0.1427	-0.1568	-0.1568	0.2574	-0.0718	0.1646	36.1640
3	1.000	6.000	0.2023	0.2900	0.5697	0.7573	-0.0915	-0.1805	0.5913	0.1808	0.2052	13.2346
4	2.000	6.000	0.2330	0.2330	0.7768	0.7768	-0.2639	-0.2639	0.8838	0.1260	0.3789	22.0729
5	3.000	6.000	0.1579	0.1579	-0.1807	-0.1807	0.0000	0.5774	0.1579	-0.1807	0.1693	0.0001
6	4.000	6.000	0.2330	0.2330	0.7768	0.7763	0.2639	0.2639	0.8838	0.1260	0.3789	-22.0730
7	5.000	6.000	0.2023	0.2900	0.5697	0.7573	0.0915	0.1805	0.5913	0.1808	0.2052	-13.2346
8	5.500	6.000	0.0428	0.0428	0.1427	0.1427	0.1568	0.1568	0.2574	-0.0718	0.1646	-36.1640
9	6.000	6.000	-0.0290	-0.0290	-0.0967	-0.0967	0.1331	0.1331	0.0745	-0.2002	0.1373	37.8661
10	0.500	5.500	0.1251	0.1251	0.2113	0.2113	-0.1651	-0.1651	0.3389	-0.0025	0.1707	37.6852
11	1.000	5.500	0.1725	0.1969	0.4434	0.4507	-0.1427	-0.1889	0.5047	0.1112	0.1967	23.2474
12	3.000	5.500	0.2413	0.2413	-0.1175	-0.1175	0.0000	0.3595	0.2413	-0.1175	0.1794	0.0001
13	5.000	5.500	0.1725	0.1969	0.4434	0.4507	0.1427	0.1889	0.5047	0.1112	0.1967	-23.2472
14	5.500	5.500	0.1251	0.1251	0.2113	0.2113	0.1651	0.1651	0.3389	-0.0025	0.1707	-37.6851
15	1.000	5.000	0.1249	0.2791	0.5926	0.6641	-0.2326	-0.3294	0.6886	0.0289	0.3298	22.4261
16	2.000	5.000	0.0701	0.0832	0.6889	0.7325	-0.2178	-0.2184	0.7579	0.0011	0.3784	17.5734
17	3.000	5.000	0.1604	0.1844	0.5893	0.6785	0.0000	0.1862	0.5893	0.1604	0.2144	-0.0001
18	4.000	5.000	0.0701	0.0832	0.6889	0.7325	0.2178	0.2184	0.7579	0.0011	0.3784	-17.5735
19	5.000	5.000	0.1249	0.2791	0.5926	0.6641	0.2326	0.3294	0.6886	0.0289	0.3298	-22.4260
20	1.500	3.500	0.0985	0.0985	1.0539	1.0539	-0.3148	-0.3148	1.1483	0.0042	0.5721	16.6915
21	3.000	4.000	0.1314	0.1314	0.8735	0.8735	0.0000	0.0073	0.8735	0.1314	0.3710	-0.0002
22	4.500	3.500	0.0985	0.0985	1.0539	1.0539	0.3148	0.3148	1.1483	0.0042	0.5721	-16.6913
23	2.000	2.000	-0.0068	-0.1789	1.4441	1.6063	-0.1286	-0.1944	1.4554	-0.0181	0.7368	5.0267
24	2.500	2.500	0.1334	0.1471	1.2348	1.3060	-0.0195	-0.0350	1.2352	0.1331	0.5510	1.0135
25	3.000	3.000	0.1820	0.2213	1.0725	1.0867	0.0000	0.0288	1.0725	0.1820	0.4452	-0.0001
26	3.500	2.500	0.1334	0.1470	1.2348	1.3060	0.0195	0.0350	1.2352	0.1331	0.5510	-1.0136
27	4.000	2.000	-0.0068	-0.1789	1.4441	1.6063	0.1286	0.1944	1.4554	-0.0181	0.7368	-5.0267
28	3.000	2.000	0.0661	0.0728	1.2610	1.2834	0.0000	0.0000	1.2610	0.0661	0.5974	-0.0001
29	1.750	1.000	-0.5687	-0.5687	1.1263	1.1263	0.1919	0.1919	1.1478	-0.5901	0.8689	-6.3809
30	4.250	1.000	-0.5687	-0.5687	1.1263	1.1263	-0.1919	-0.1919	1.1478	-0.5901	0.8689	6.3808
31	1.500	0.0	-0.5696	-0.5696	0.7756	0.7756	0.4820	0.4820	0.9305	-0.7244	0.8275	-17.8117
32	3.000	0.0	-0.4190	-0.4190	0.9453	0.9453	0.0000	0.0000	0.9453	-0.4190	0.6822	-0.0001
33	4.500	0.0	-0.5696	-0.5696	0.7757	0.7757	-0.4820	-0.4820	0.9305	-0.7244	0.8275	17.8113

STRESSES AT QUADRILATERAL CENTROIDS (POINT NO. = 33 + ELEMENT NO.)

35	2.000	5.500	0.0210	1.1787	-0.2954	1.2497	-0.0500	0.6498	13.5190
36	4.000	5.500	0.0210	1.1787	0.2954	1.2497	-0.0500	0.6498	-13.5190
38	2.167	3.833	0.0849	0.9752	-0.1852	1.0122	0.0479	0.4821	11.2963
39	3.833	3.833	0.0849	0.9751	0.1852	1.0121	0.0479	0.4821	-11.2965
41	3.000	0.933	-0.6816	1.4006	0.0000	1.4006	-0.6816	1.0411	-0.0001

SIGMA XX



CONTOUR LINE REFERENCE VALUES

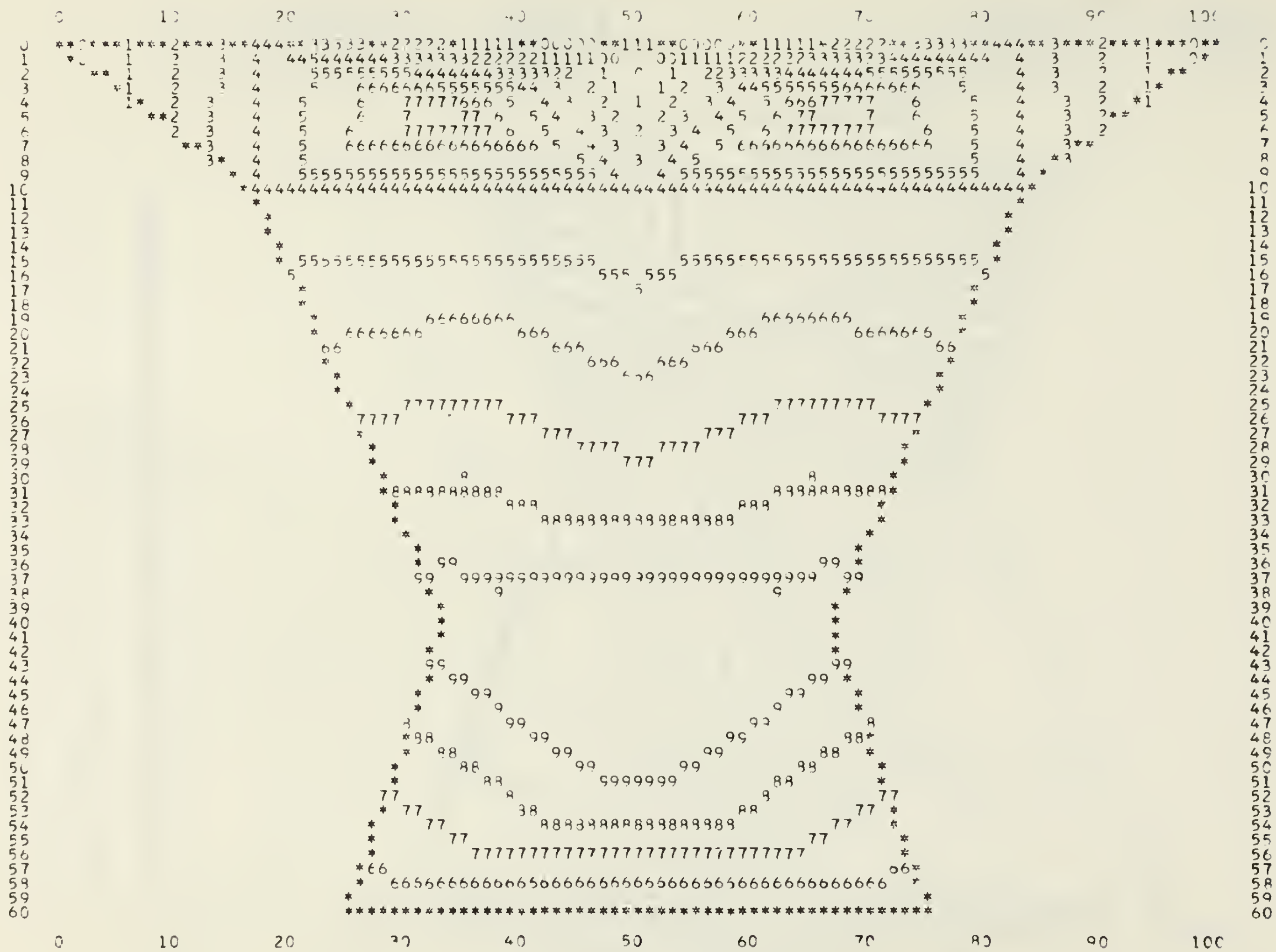
C =	0.0	OR	-0.0	1 =	6.00000E-02	OR	-6.00000E-02
2 =	1.20000E-01	OR	-1.20000E-01	3 =	1.80000E-01	OR	-1.80000E-01
4 =	2.40000E-01	OR	-2.40000E-01	5 =	3.00000E-01	OR	-3.00000E-01
6 =	3.60000E-01	OR	-3.60000E-01	7 =	4.20000E-01	OR	-4.20000E-01
8 =	4.80000E-01	OR	-4.80000E-01	9 =	5.40000E-01	OR	-5.40000E-01
D =	6.00000E-01	OR	-6.00000E-01				

(*-DENOTES BOUNDARY POINTS)

Note: All figures are scaled in the Y-direction so that a body with a square envelope will plot a figure that fits a standard 8½ x 11 sheet.

When the body length/height ratio exceeds 1, the figure is plotted in a 100x100 line perimeter.

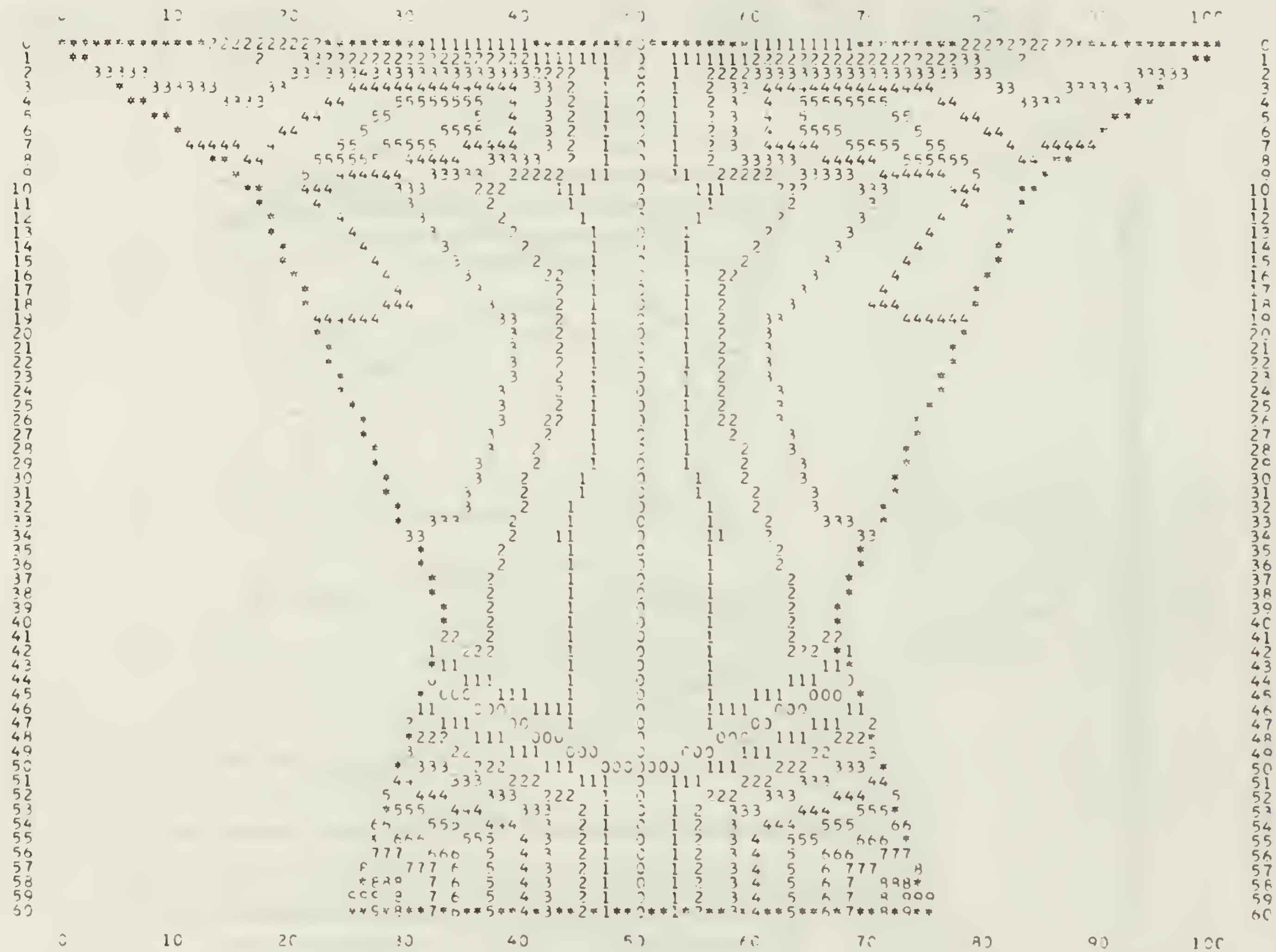
SIGMA YY



CONTOUR LINE REFERENCE VALUES

0 =	0.0	OR	-0.0		1 =	1.50000E-01	OR	-1.50000E-01
2 =	3.00000E-01	OR	-3.00000E-01		3 =	4.50000E-01	OR	-4.50000E-01
4 =	5.99999E-01	OR	-5.99999E-01		5 =	7.49999E-01	OR	-7.49999E-01
6 =	8.99999E-01	OR	-8.99999E-01		7 =	1.05000E 00	OR	-1.05000E 00
8 =	1.20000E 00	OR	-1.20000E 00		9 =	1.35000E 00	OR	-1.35000E 00
0 =	1.50000E 00	OR	-1.50000E 00			(*-DENOTES BOUNDRY POINTS)		

TAU XY

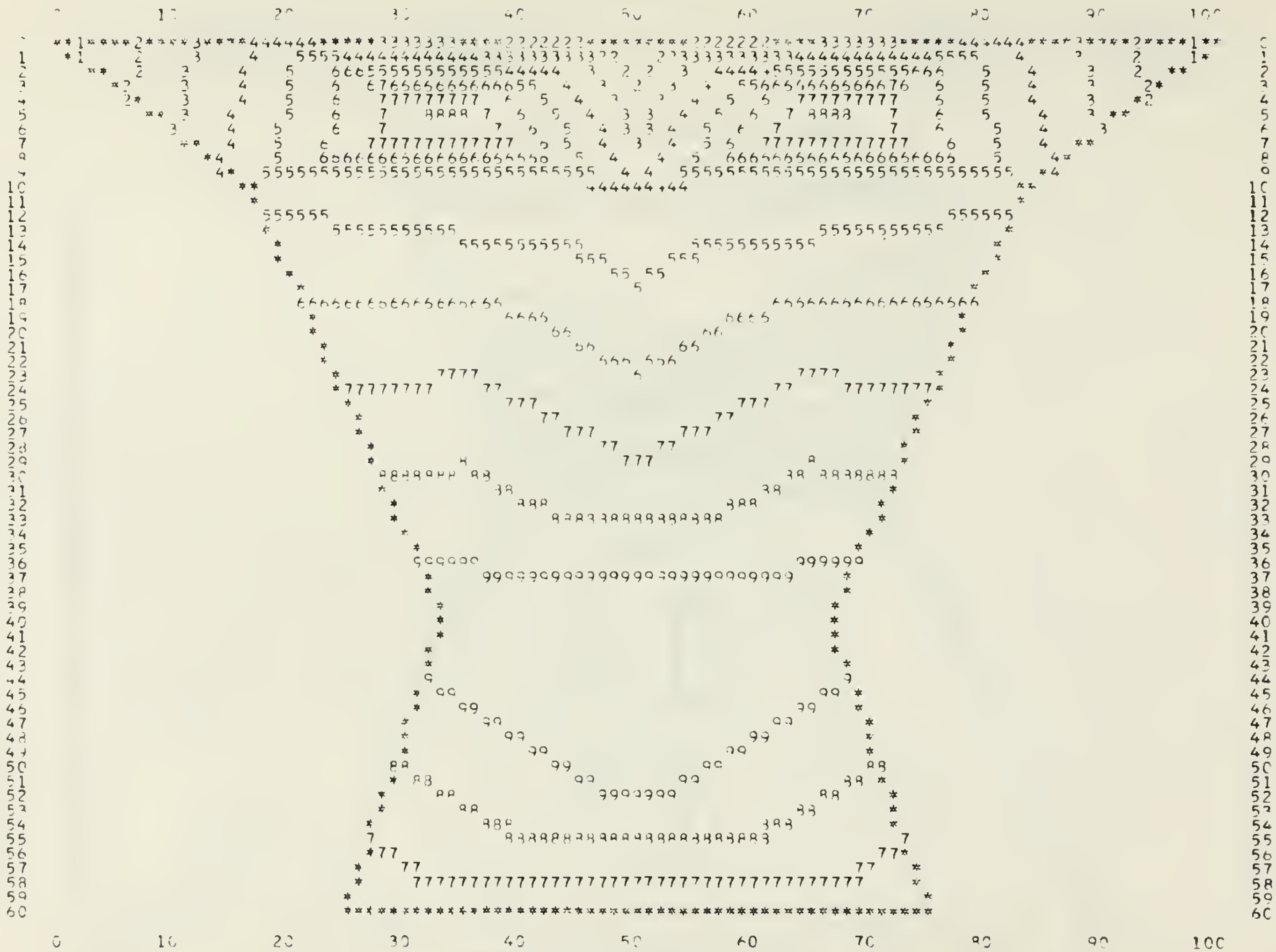


CONTOUR LINE REFERENCE VALUES

0 =	0.0	OR	-0.0		1 =	5.00000E-02	OR	-5.00000E-02
2 =	9.99999E-02	OR	-9.99999E-02		3 =	1.50000E-01	OR	-1.50000E-01
4 =	2.00000E-01	OR	-2.00000E-01		5 =	2.50000E-01	OR	-2.50000E-01
6 =	3.00000E-01	OR	-3.00000E-01		7 =	3.50000E-01	OR	-3.50000E-01
8 =	4.00000E-01	OR	-4.00000E-01		9 =	4.50000E-01	OR	-4.50000E-01
D =	5.00000E-01	OR	-5.00000E-01					

(*-DENOTES BOUNDRY POINTS)

SIGMA MAX

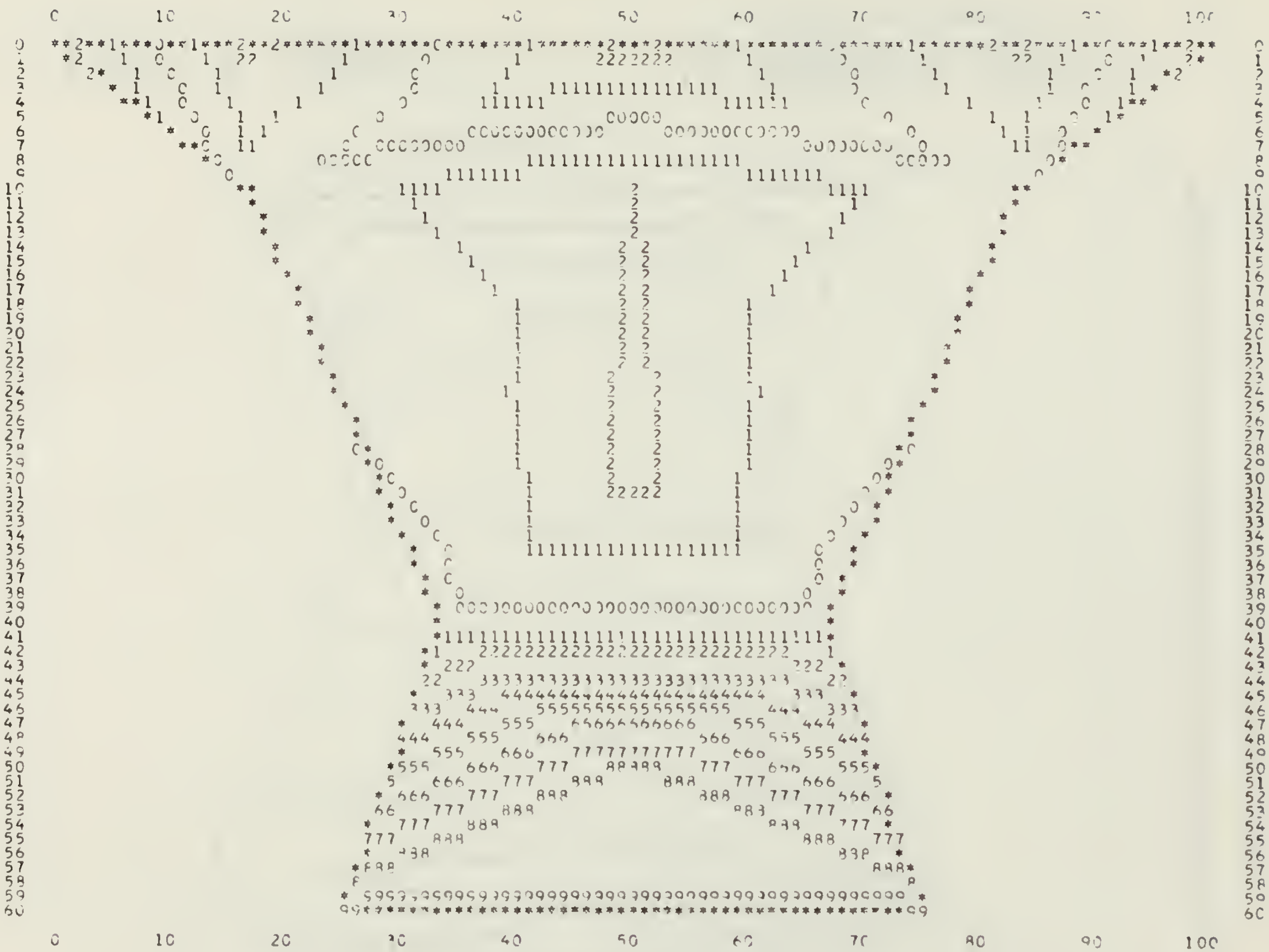


CONTOUR LINE REFERENCE VALUES

0 =	0.0	OR	-0.0	1 =	1.50000E-01	OR	-1.50000E-01
2 =	3.00000E-01	OR	-3.00000E-01	3 =	4.50000E-01	OR	-4.50000E-01
4 =	5.99999E-01	OR	-5.99999E-01	5 =	7.49999E-01	OR	-7.49999E-01
6 =	8.99999E-01	OR	-8.99999E-01	7 =	1.05000E 00	OR	-1.05000E 00
8 =	1.20000E 00	OR	-1.20000E 00	9 =	1.35000E 00	OR	-1.35000E 00
0 =	1.50000E 00	OR	-1.50000E 00				

(*-DENOTES BOUNDRY POINTS)

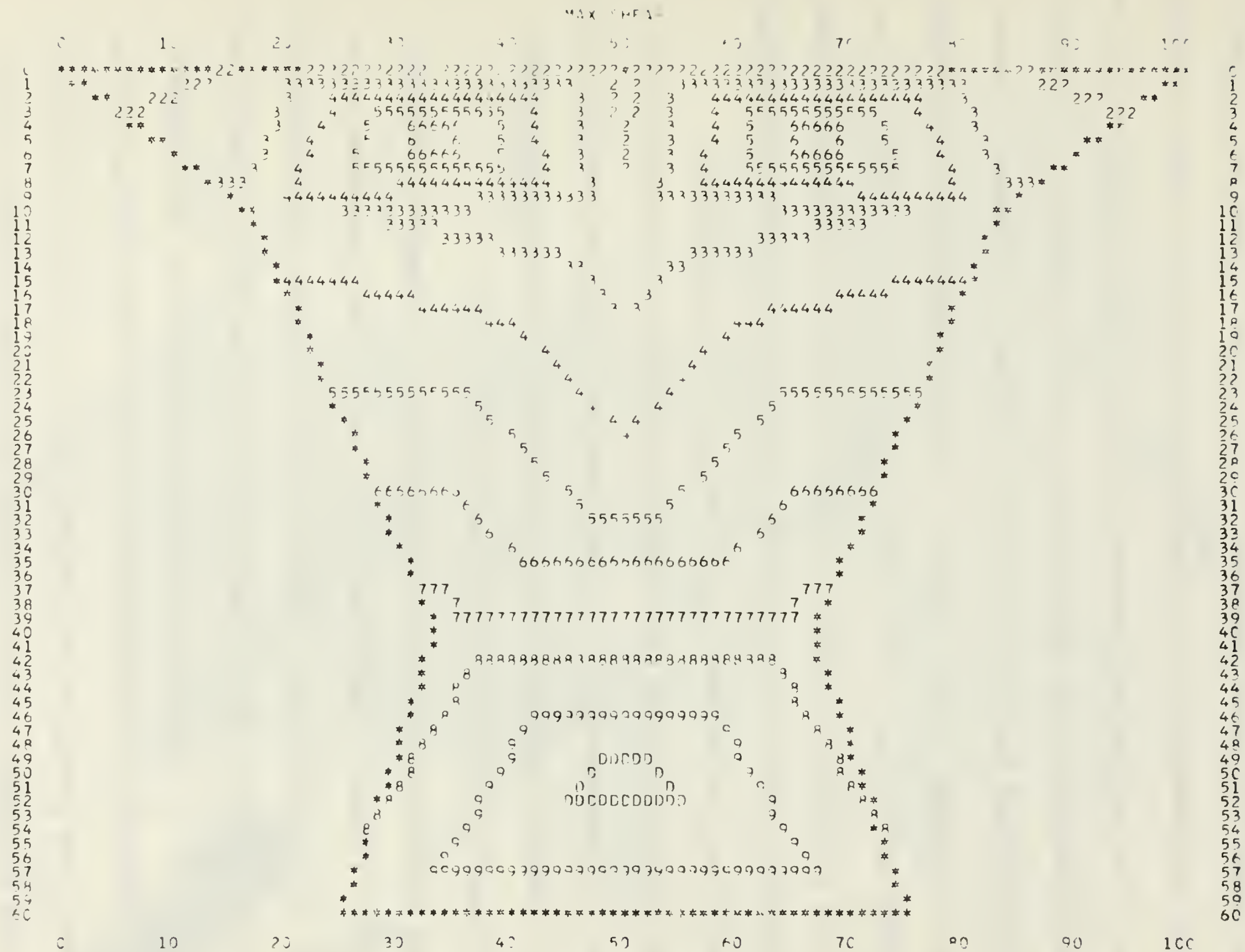
SIGMA MIN



CONTOUR LINE REFERENCE VALUES

0 =	0.0	OR	-0.0	1 =	7.99999E-02	OR	-7.99999E-02
2 =	1.60000E-01	OR	-1.60000E-01	3 =	2.40000E-01	OR	-2.40000E-01
4 =	3.20000E-01	OR	-3.20000E-01	5 =	4.00000E-01	OR	-4.00000E-01
6 =	4.80000E-01	OR	-4.80000E-01	7 =	5.59999E-01	OR	-5.59999E-01
8 =	6.39999E-01	OR	-6.39999E-01	9 =	7.19999E-01	OR	-7.19999E-01
0 =	7.99999E-01	OR	-7.99999E-01				

(*-DENOTES BOUNDRY POINTS)



CONTOUR LINE REFERENCE VALUES

```
C =   0.0           OR   -0.0               1 =   9.99999E-02   OR   -9.99999E-02
2 =   2.00000E-01   OR   -2.00000E-01       3 =   3.00000E-01   OR   -3.00000E-01
4 =   4.00000E-01   OR   -4.00000E-01       5 =   5.00000E-01   OR   -5.00000E-01
6 =   5.99999E-01   OR   -5.99999E-01       7 =   6.99999E-01   OR   -6.99999E-01
8 =   7.99999E-01   OR   -7.99999E-01       9 =   8.99999E-01   OR   -8.99999E-01
D =   9.99999E-01   OR   -9.99999E-01       (*-DENOTES BOUNDRY POINTS)
```

EXECUTION TIME FOR ** CNTPLT= 9.794 SECONDS **

APPENDIX 2 Practical Aspects of the Computer Alogrithm

This appendix presents practical aspects of the computer alogrithm for the direct stiffness procedures, equation solver, and the application of boundary conditions.

The direct stiffness procedure is the process whereby the complete structure stiffness matrix is formed from the individual element stiffness matrices. The process is conceptually simple, since it requires only the systematic addition of the stiffnesses of all the elements in the system. Difficulties arise when the concept is transposed to a computer process where storage space is at a premium. A firm understanding of the methods of assembly and storage of the complete stiffness matrix within the program is of importance for those desiring to modify the existing program. These methods control the programming techniques for application of boundary conditions, and the solution of the equilibrium equations.

The superposition of each element stiffness, in the formation of the complete stiffness matrix, is accomplished by adding its individual terms into the complete stiffness matrix according to the nodal point number of the elements. This is illustrated in Figure A2.1.

The program could theoretically construct the complete stiffness matrix in the manner illustrated in Fig. A2.1; a two-dimensional array could be defined and simple indexing methods utilized to superpose the element stiffnesses. The storage requirements for this straightforward approach would be a square array with side dimension of (2 x Maximum Number of Nodes). The resulting array (2100 x 2100) would occupy the total main core storage of 70 computers of the size used for this program. The techniques used to alleviate the storage problem utilizes three characteristics of the complete stiffness matrix.

The complete stiffness matrix is

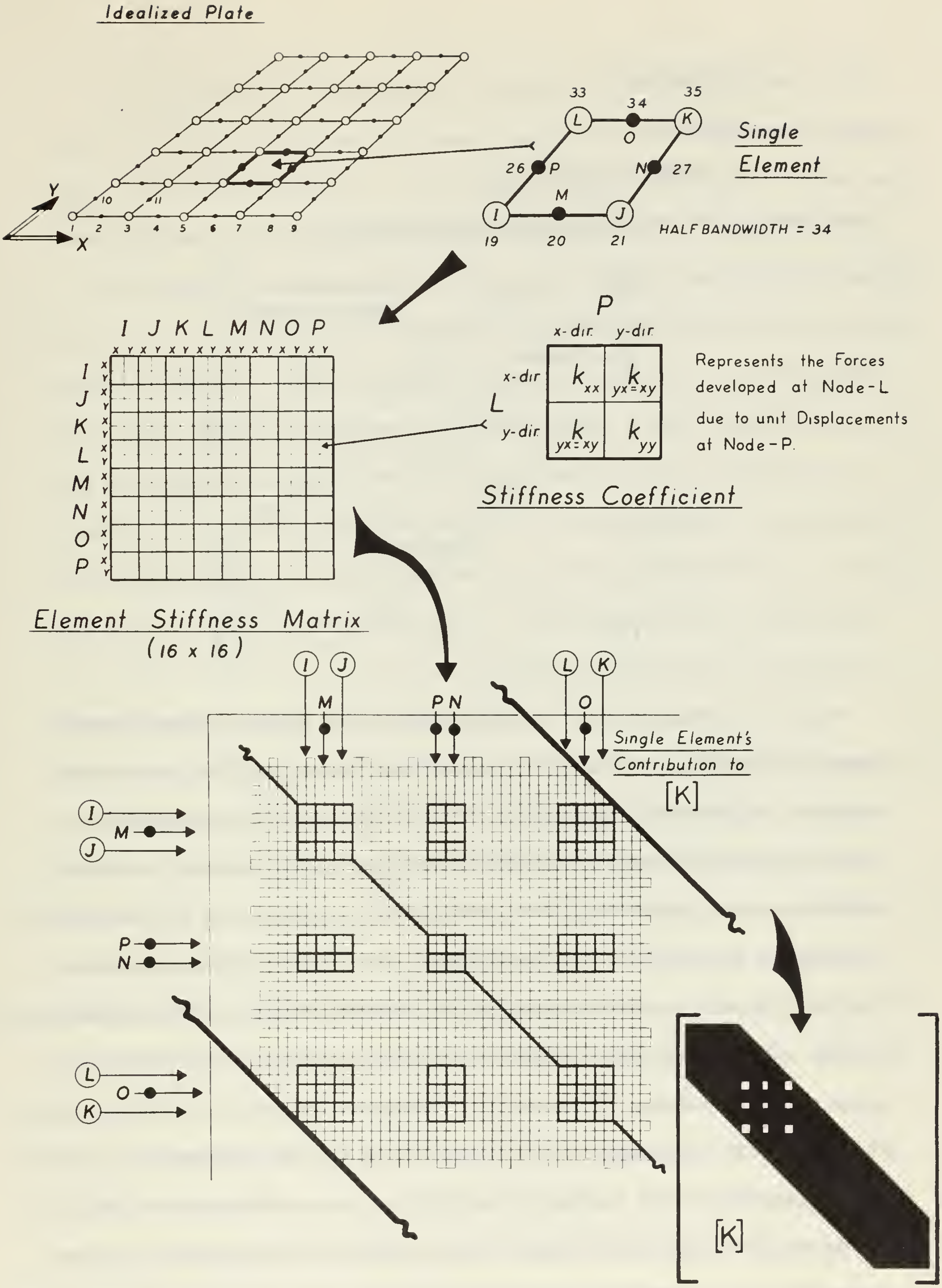


FIG. A2.1 DIRECT STIFFNESS PROCEDURE

(1) Symmetric

(2) Banded

(3) Sparsely populated

These properties are illustrated in Figure A2.2.

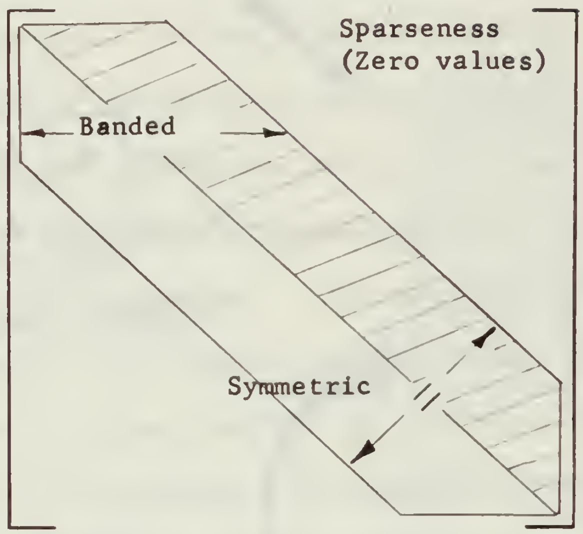


Figure A2.2 Complete Stiffness Matrix Characteristics

Symmetry permits a reduction of approximately one-half in the storage required, since only a triangular half of the matrix is required to retain significant data. The banded and sparseness properties allow further savings, since only data between the diagonal and the band limits is non-zero and required in computation. The cross-hatched portion of Fig. A2.2 shows the actual quantity of storage required. The storage space problem is only partially solved, since space approximately equal to an array size dimensioned (½ Bandwidth x Diagonal Length) is still required. In the program the maximum half-bandwidth is 160 and the maximum diagonal length (equals a side dimension) is 2100. The resulting (160 x 2100) array is small enough to be stored on an external disk storage unit, but cannot be housed completely in main core (5 would be required) where high speed calculations, independent of internal/external storage transfer time, can be performed.

Considerations of the equation solving method and transfer time reduction dictate the final techniques used in the storage and manipulation of the data contained in the complete stiffness matrix. These two considerations will be discussed before proceeding.

The equation solver used in the program employs Cholesky's algorithm [4], [10]. The fundamental concept of the solution process is to perform operations on the complete stiffness matrix to reduce it to triangular form, after which the unknown displacements are found by back substitution procedures. The limited coupling characteristic of the set of equations allows the reduction in form to be accomplished in a manner where a subgrouping of equations can be reduced independently in a process termed triangular decomposition. The decomposition process does not modify the load vector in the equilibrium equations, and once completed, the decomposed form of the complete stiffness matrix may be stored for possible use with additional load cases. This accounts for the greatly reduced running time for additional load cases. The subgroups of equations are handled as individual blocks of equations. Figure A2.3 shows a block structure used for decomposing the complete stiffness matrix. The figure also illustrates two points concerning block structure; (1) element stiffness contributions may enter two different blocks, and (2) in order to maintain equal block size, the first must contain a zero value area.

The transfer time between external and internal core storage is a function of the data address determination time and the physical time required to transfer the data. Where many individual units or small groups of data are being moved from within a larger data set (the situation in our case), the address computation time is usually the largest contributor to the overall time. To reduce address time to a minimum,

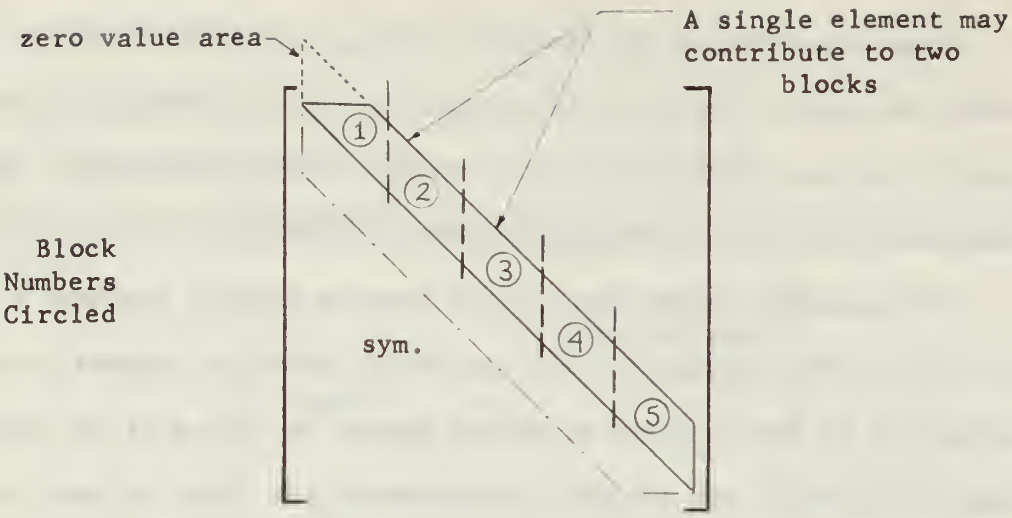


Figure A2.3 Block Structure used for decomposing [K]

the use of singly dimensioned arrays, with their attendent short address calculations, is required.

Returning to the complete stiffness matrix, with its maximum data storage requirement of a (160 x 2100) array, the considerations of the preceeding paragraphs are applied. The final storage technique for the resulting matrix data calls for data in block form with each block stored individually on an external disk unit. Each block is transferred, as required, to main core storage for the formation, modification, or solution procedures. Each block is stored in a one-dimensional array form. The overall storage method is illustrated in Figure A2.4, where an original complete stiffness matrix is shown, along with a virtual array depicting the actual data that requires storage, and the final one-dimensional array form of one of the stored blocks. Block divisions are indicated in the figure. The rows of the virtual array, and the order of data in the one-dimensional array, are the columns (read bottom to top) of the upper half-bandwidth of the complete stiffness matrix. The first

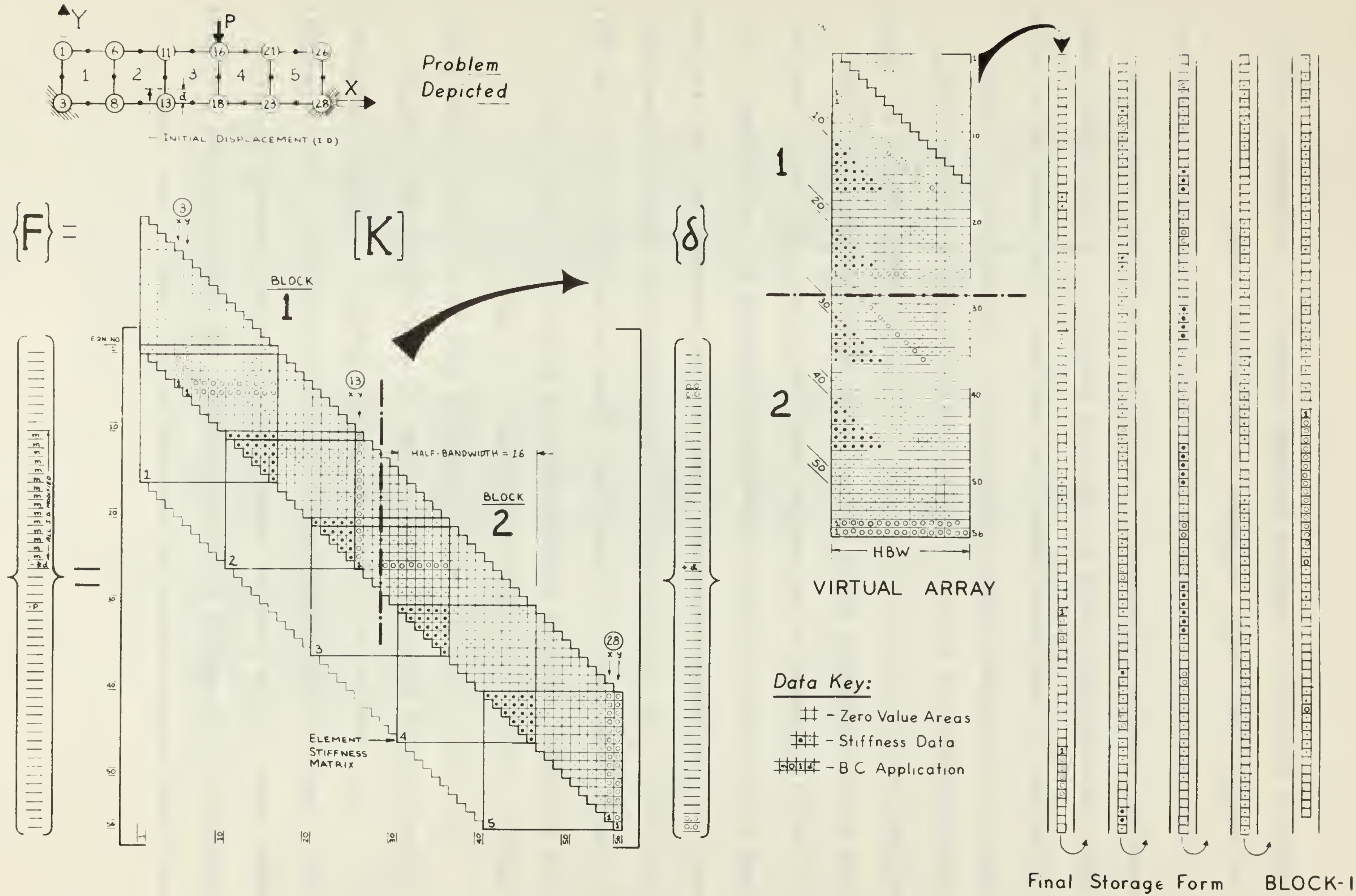


FIG. A2.4 COMPLETE STIFFNESS MATRIX STORAGE METHOD AND BOUNDARY CONDITION APPLICATION column of the virtual array is the main diagonal of the complete matrix. The original rows become diagonals in the virtual array.

The application of boundary conditions to the structure equilibrium equations involves adjustment of the complete stiffness matrix to reflect the constrained nodes or initially displaced and constrained nodes. In the case of constrained nodes, the condition is accounted for by striking out the rows and columns corresponding to the degrees of freedom associated with the boundary condition and replacing the corresponding diagonal terms with a non-zero value. If initial displacements are specified the corresponding column of the matrix must be multiplied by the initial displacement value, and the resulting vector subtracted from the load vector before the above constraint procedures are carried out. In addition, the known displacement value must be inserted into the displacement vector. The boundary condition application procedure is illustrated in Figure A2.4, where a 5-element bar, fixed at the ends (nodes 3 and 28); with an initial positive Y-displacement at node 13, is loaded with a negative Y-direction concentrated force, P, at node 16.

The carry over of boundary conditions into the virtual storage array and one block of the final storage form, is also illustrated in Figure A2.4. Since the stiffness data is actually stored in a singly dimensioned array, special indexing and address accounting procedures (see subroutine) FORMK) must be used to allow complete row and column boundary condition procedures to be applied in situations where two data blocks are involved; each block being available for manipulation only when in main core.

```
C
C
C
C     ************************************************************************
C     PSE-LST -  LINEAR ELASTIC ANALYSIS OF PLANE STRESS OR PLANE
C     STRAIN PROBLEMS USING LINEAR STRAIN TRIANGLES (LST)
C     THE BASIC MESH INPUT UNIT IS A QUADRILATERAL FORMED WITH
C     FOUR LST, HAVING 16 EXTERNAL (FUNDAMENTAL) DEGREES OF FREEDOM
C     AND 10 INTERNAL (ADDITIONAL) DEGREES OF FREEDOM
C     ************************************************************************
C     PROGRAMMER:  CARLOS A. FELIPPA,  JUNE 1966
C     REVISED AND ADOPTED TO IBM OS/360    JOHN P. MALONE  JULY 1968
C
C
C     MAIN
C
C
      COMMON
     1 NUMEL, NUMCP, NUMNP, NUMBC, NLOAD, MAXIT, NEQ,
     2 IBANDW, NEQBC, NRUN, NTRA, NSKEWB, LNQ(6,4),
     3 T1, T2, T3, T4, T5, THERL
      LOGICAL PGRAPH, T1, T2, T3, T4, T5
      COMMON /CMATPR/  YM(6), PR(6), RHO(6), ALFA(6)
      COMMON /CELMAR/  NP(350,8), NEBC(250), BANGLE(250), DISPBC(250)
      COMMON /CNTARG/  NUMPB, NSK, PGRAPH, IGRTAG(6), SPACNG(6),
     2  GRHEAD(3,6), NPB(50), NELSKP(50)
C
  100 CALL SETUP
  200 CALL LDINPT
      IF(NRUN.LE.1) CALL FORMK
  300 CALL SOLVE
      CALL STRESS
      IF (PGRAPH)  CALL CNTPLT
      NRUN = NRUN + 1
      IF (NRUN.LE.NLOAD)  GO TO 200
      IF (T5)  GO TO 100
      STOP
      END
```

```
      SUBROUTINE WRDISK(NTRACK, A,NCT)
C
C
      INTEGER LAST/0/
      REAL*8 NAME(2)/'WRDISK','RDDISK'/
      DIMENSION  A(1)
      DEFINE FILE 7(20000,184,E,I)
C
C
      IF(NTRACK .GT. 1999) GO TO 900
C
C          SET UP TO WRITE ON NEXT FREE AREA.
C
      IF(NTRACK .LT. 0) NTRACK =(LAST+9)/10
      N = NTRACK
      N = N*10 + 1
      IF (NCT.GT. 46)GO TO 50
      WRITE (7'N,1000) (A(J),J=1,NCT)
      IF (LAST .LT. I) LAST = I
      IF(LAST.GT. 19999)LAST=0
C
      RETURN
C
C       WRITE OUT RECORDS FOR COUNTS IN EXCESS OF 46 WORDS.
C
   50 JI = 47
      WRITE (7'N,1000) (A(J),J=1,46)
   75 JE = JI + 45
      IF ( JE .GE. NCT) GO TO 100
      WRITE (7'I,1000) (A(J),J=JI,JE)
      JI = JI+46
      GO TO 75
  100 WRITE (7'I,1000) (A(J),J=JI,NCT)
      IF(I.GT.LAST) LAST=I
      IF(LAST.GT. 19999)LAST=0
C
      RETURN
C
C
C
      ENTRY RDDISK(NTRACK,B,NCT)
      DIMENSION B(1)
C
      IF(NTRACK .GT. 1999) GO TO 910
      N=NTRACK*10 + 1
      IF (NCT .GT.46) GO TO 150
      READ (7'N,1000) (B(J), J=1,NCT)
```

```
C
      RETURN
C
  150 READ (7'N,1CCC) (B(J),J=1,46)
      JI = 47
  175 JE = JI + 45
      IF (JE .GE. NCT) GO TO 20C
      READ  (7'I,1CCO) (B(J),J=JI,JE)
      JI = JI + 46
      GO TO 175
  200 READ  (7'I,1CCO) (B(J),J=JI,NCT)
C
      RETURN
C
C
 1000 FORMAT (46A4)
  900 K=1
      GO TO 920
  91C K=2
C
  920 WRITE(6,192C)NAME(K),NTRACK
 1920 FORMAT('0 ERRCR IN CALL OF ',A6,', NTRACK TOO LARGE,(MUST BE .LT.
     1THAN 2000)' / '  NTRACK =',I10)
      STCP
      END
```

```
      SUBROUTINE SETUP
C
C
C
C     ******************************************************************
C     SETUP INPUTS DATA AND EVALUTES ELEMENT STIFFNESSES
C     ******************************************************************
C
      COMMON
     1 NUMEL, NUMCP, NUMNP, NUMBC, NLOAD,MAXIT,NEQ,
     2 IBANDW, NEQBC, NRUN, NTRA, NSKEWB, LNQ(6,4),
     3 T1, T2,T3, T4, T5, THERL
      COMMON /CMATPR/  YM(6), PR(6), RHO(6), ALFA(6)
      COMMON /CELMAR/  NP(350,8), NEBC(250), BANGLE(250), DISPBC(250)
      COMMON /QUADRG/ S11(16,16), S21(10,26), X(5),Y(5), E, XU, THICK
      COMMON /STFARG/ ST(16,16), B(3), A(3), AREA, ET, NU, THIK
      COMMON /CNTARG/ NUMPB, NSK, PGRAPH, IGRTAG(6), SPACNG(6),
     2  GRHEAD(3,6), NPB(50), NELSKP(50)
      DIMENSION
     1 XORD(1050), YORD(1050), XCENT(350), YCENT(350),
     2 MAT(350), TH(350), HEAD(20), GRTITL(3,6), IPERM(3),
     3 IPERM4(4), QUDR(3), TRNG(3), DUM(1)
      EQUIVALENCE (DUM,ST)
      REAL NU
      REAL*8  FLAG, CHECK
      LOGICAL T1,T2,T3,T4, T5,QD
      DATA IPERM /2,3,1/, IPERM4/2,3,4,1/
      DATA  FLAG /8HSTART   /, TEST /ZFFFFF001/
      DATA QUDR(1) /12H QUAD'LTERAL/, TRNG(1) /12H TRIANGLE   /
      DATA GRTITL(1,1) /' SIGMA XX     SIGMA YY     TAU XY       SIGMA MAX
     1  SIGMA MIN    MAX SHEAR  '/
C
C     ******************************************************************
C     INITIALIZATION
C     ******************************************************************
C
  100 NRUN = 1
      IFLAG = 0
      MAXEL = 350
      MAXNP = 1050
      MAXPD = 80
      MAXBC = 250
      DO 110  I = 1,4
      J = IPERM4(I)
      LNQ(1,I) = I
      LNQ(2,I) = J
      LNQ(3,I) = 9
```

```
      LNQ(4,I) = I + 4
      LNQ(5,I) = J + 9
  110 LNQ(6,I) = I + 9
      DO 120  I = 1,3
      DO 120 J = 1,6
  120 GRHEAD(I,J) = GRTITL(I,J)
C
C     ******************************************************************
C     READ AND PRINT OF INPUT DATA
C     ******************************************************************
C
  140 READ(5,9,ERR=1060)  CHECK
      IF (CHECK.NE.FLAG)  GO TO 140
      READ(5, 10)  HEAD
      PRINT 11, HEAD
    9 FORMAT (A8)
   10 FORMAT  (20A4)
   11 FORMAT  (1H1, 20A4)
C
C     CONTROL PARAMETERS
C
      READ(5, 15)  NUMEL, NUMCP, NUMNP, NUMBC, NUMPB, NLOAD, NMAT,
     1 MAXIT, T1, T2, T3, T4, T5
      PRINT 16, NUMEL, NUMCP, NUMNP, NUMBC, NUMPB, NLOAD, NMAT,
     1 MAXIT, T1, T2, T3, T4, T5
   15 FORMAT (8I4, 5L2)
   16 FORMAT (//
     1 35H NUMBER OF ELEMENTS  . . . . . . . .      I8 /
     2 35H NUMBER OF CORNER POINTS . . . . . .      I8 /
     3 35H NUMBER OF NODAL POINTS  . . . . . .      I8 /
     4 35H NUMBER OF BOUNDARY CONDITIONS . .       I8 /
     5 35H NUMBER OF DEF. BOUNDARY POINTS  .       I8 /
     6 35H NUMBER OF LOAD CASES. . . . . . . .     I8 /
     7 35H NUMBER OF DIFFERENT MATERIALS . .       I8 /
     8 35H MAX NO OF RESIDUAL ITERATIONS . .       I8 //
     9 35H FLAG EQUAL TYPE QUADRILATERALS. .       L8 /
     1 35H FLAG PUNCH DISPLACEMENTS. . . . .       L8 /
     2 35H FLAG PUNCH STRESSES . . . . . . . .     L8 /
     3 35H FLAG PRINT ELEMENT STRESSES . . .       L8 /
     4 35H FLAG NEW JOB FOLLOWS  . . . . . . .     L8 )
C
C     MATERIAL PROPERTIES
C
      READ(5, 20)  (I, YM(I), PR(I), RHO(I), ALFA(I),  L=1,NMAT)
      PRINT 22, (I,YM(I), PR(I), RHO(I), ALFA(I), I=1,NMAT)
   22 FORMAT (20H-MATERIAL PROPERTIES //9H MAT. NO., 5X,
     1  15HELASTIC MODULUS, 5X,  15HPOISSON'S RATIO, 11X, 7HDENSITY,
```

```
     2 6X, 17HDILATATICN COEFF.   //(I9,2P1E20.5,0P2F19.5,1P1E22.5) )
  20 FORMAT  (I4, 4E10.3)
C
C     BCUNDRY POINTS
C
      READ(5, 24)  (NPB(I),  I=1,NUMPB)
      PRINT 26, (NPB(I),  I=1,NUMPB)
  24  FORMAT  (20I4)
  26  FORMAT  (25H-DEFINING BOUNDARY POINTS  //(20I5))
C
C     ELEMENT ARRAY
C
      READ(5, 30)  (N, (NP(N,I), I=1,8), MAT(N), TH(N),  L=1,NUMEL)
      PRINT 31
      DO 150  N = 1,NUMEL
      IF (MAT(N).LE.0)  MAT(N) = 1
      IF (TH(N).LE.C.)  TH(N) = 1.
      QD = NP(N,7).GT.0
      IF (QD)          PRINT 32, QUDR,N,(NP(N,I),I=1,8),MAT(N),TH(N)
 150  IF (.NOT.QD)    PRINT 33, TRNG,N,(NP(N,I),I=1,6),MAT(N),TH(N)
  30  FORMAT  (10I4,F10.3)
  31  FORMAT  (14H1ELEMENT ARRAY  //4X, 7HELEMENT, 10X, 1HI, 5X, 1HJ,
     1 5X, 1HK, 5X, 1HL, 5X, 1HM, 5X, 1HN, 5X, 1HO, 5X, 1HP,
     2 4X, 8HMAT.TYPE, 5X,9HTHICKNESS /1X)
  32  FORMAT  (3A4, I4, 8I6, I11, F14.4)
  33  FORMAT  (3A4, I4, 6I6, 12X, I11, F14.4)
C
C     NODAL PCINT CCORDINATES
C
      DO 200 N = 1,NUMNP
      XORD(N) = TEST
 200  YORD(N) = TEST
      PRINT 35
  35  FORMAT(1H1,'CORNER PCINT COORDINATES',//,
     1 3X, 2(' PCINT          X-ORD           Y-ORD',7X),/,1X)
      DO 220 M = 1,NUMCP
C
C     COORDINATE ARRAY
C
      READ(5, 36)  N, XORD(N), YORD(N)
      IF(M-((M/2)*2)) 210,210,205
 205  WRITE(6,38) N, XORD(N), YORD(N)
      GO TO 220
 210  WRITE(6,39) N, XORD(N), YORD(N)
 220  CONTINUE
  36  FORMAT  (I4, 2F8.4)
  38  FORMAT(1H ,1X,I6,1X,2F12.4)
```

```
   39 FORMAT(1H+,T40,I6,1X,2F12.4)
C
C     BOUNDARY CONDITIONS
C
      DO 240 N = 1,MAXBC
      DISPBC(N) = 0.0
  240 BANGLE(N) = 0.
      J = 0
      NSKEWB = 0
      PRINT 45
   45 FORMAT(1H1,' BOUNDARY CONDITIONS',//,' NODAL     TAG     BOUNDARY AN
     1GLE     INIT. BOUNDARY',/,' POINT    VALUE    (TAG=0, X-DISP)    DISPL
     2ACEMENT',/,1X)
      DO 250  N = 1,NUMBC
C
C     BOUNDRY CONDITION ARRAY
C
      READ(5, 50)  M, L, ANGLE, DISP
      PRINT 52, M, L, ANGLE, DISP
      K2 = 2*M
      K1 = K2 - 1
      IF (L-1)  242,244,246
  242 J = J + 2
      NEBC(J) = K2
      NEBC(J-1) = K1
      DISPBC(J) = DISP
      DISPBC(J-1) = ANGLE
      GO TO 250
  244 J = J + 1
      NEBC(J) = K1
      DISPBC(J) = DISP
      GO TO 250
  246 J = J + 1
      NEBC(J) = K2
      IF (ANGLE.NE.0.)  NSKEWB = 1
      BANGLE(J) = ANGLE/57.29578
      DISPBC(J) = DISP
  250 CONTINUE
   50 FORMAT (2I4, 2E15.5)
   52 FORMAT (2I6, 2F20.10)
      NEQBC = J
      NEQ = 2*NUMNP
      IF (NUMEL.GT.MAXEL)  GO TO 1000
  260 IF (NUMNP.GT.MAXNP)  GO TO 1010
  270 IF (NEQBC.GT.MAXBC)  GO TO 1020
C
C     ***********************************************************************
```

```
C      DETERMINATICN CF BANC WIDTH
C      ********************************************************************
C
  28C   K = C
       DO 320  N = 1,NUMEL
       DO 320  I = 1,7
       K1 = NP(N,I)
       IF  (K1.LE.0)  GO TC 320
       II = I + 1
       DO 300  J = II,8
       K2 = NP(N,J)
       IF (K2.LE.0)  GC TO 300
       M = IABS(K2-K1)
       IF  (M.GT.K)  K = M
       IF (M.LE.MAXPC)     GC TO 300
       PRINT 6C, MAXPD, N
  300 CONTINUE
  320 CONTINUE
       IBANDW = 2*K + 2
       PRINT 62, IBANDW
   60 FORMAT (31HOMAX. NODAL POINT DIFFERENCE OF I5,
     1 21H  EXCEEDEC, ELEMENT = I5)
   62 FORMAT (//, 1X, 'HALF-BANDWIDTH = ',I4)
C
C      ********************************************************************
C      COMPUTATION OF CENTROID COORDINATES FOR QUADRILATERALS
C      ANC CHECK FCR INPUT MESH ERRORS
C      ********************************************************************
C
       DO 400 N = 1,NUMEL
       IF (NP(N,7).LE.0)  GC TC 360
       DO 340  I = 1,4
       J = IPERM4(I)
       K1 = NP(N,I)
       K2 = NP(N,J)
       X(I) = XORD(K1)
       Y(I) = YORD(K1)
       IF  (X(I).EQ.TEST.OR.Y(I).EQ.TEST)  GO TO 1040
       M = NP(N,I+4)
       XORD(M) = 0.5*(XORD(K1)+XORD(K2))
  340 YORD(M)  = 0.5*(YORD(K1)+YORD(K2))
       X1 = X(1)
       Y1 = Y(1)
       X3 = X(3)
       Y3 = Y(3)
       X24 = X(2) + X(4)
       Y24 = Y(2) + Y(4)
```

```
      X124 = (X1+X24)/3.
      X234 = (X3+X24)/3.
      Y124 = (Y1+Y24)/3.
      Y234 = (Y3+Y24)/3.
      A124 = (X(2)-X1)*(Y(4)-Y1) - (X(4)-X1)*(Y(2)-Y1)
      A234 = (X(4)-X3)*(Y(2)-Y3) - (X(2)-X3)*(Y(4)-Y3)
      AREA = A124 + A234
      IF (A124.LT.C..OR.A234.LT.0.)  GO TO 1050
      R1 = A124/AREA
      R2 = A234/AREA
      XCENT(N) = X124*R1 + X234*R2
      YCENT(N) = Y124*R1 + Y234*R2
      GO TO 400
  360 DO 380  I = 1,3
      J = IPERM(I)
      K1 = NP(N,I)
      K2 = NP(N,J)
      X(I) = XORD(K1)
      Y(I) = YORD(K1)
      IF (X(I).EQ.TEST.CR.Y(I).EQ.TEST)  GO TO 1040
      M = NP(N,I+3)
      XORD(M) = 0.5*(XORD(K1)+XORD(K2))
  380 YORD(M) = 0.5*(YORD(K1)+YORD(K2))
      A2 = X(3)-X(1)
      A3 = X(2)-X(1)
      B2 = Y(3)-Y(1)
      B3 = Y(2)-Y(1)
      AREA = A3*B2-B3*A2
      IF  (AREA.LT.C.)  GO TO 1050
  400 CONTINUE
      IF (IFLAG.NE.C)   STOP
      IF  (IFLAG.EQ.0)  GO TO 420
      IF (.NOT.T5)  STOP
      GO TO 100
  420 REWIND 8
      WRITE (8)  (XORD(N),N=1,NUMNP),(XCENT(N),N=1,NUMEL),
     1           (YORD(N),N=1,NUMNP),(YCENT(N),N=1,NUMEL)
      WRITE (8) (MAT(N),N=1,NUMEL)
      WRITE (8) (TH(N), N=1,NUMEL)
C
C     ****************************************************************
C     COMPUTATION OF ELEMENT STIFFNESSES
C     ****************************************************************
C
      L = 0
      NTRACK = 0
      DO 450  I = 1,256
```

```
  450 DUM(I) = 0.
      DO 700  N = 1,NUMEL
      M = MAT(N)
      IF  (NP(N,7).LE.C)  GC TO 600
C
C     CUACRILATERAL
C
      IF  (T1.ANC.L.GT.C)  GO TO 550
      L = L + 1
      DC 520  I = 1,4
      K = NP(N,I)
      X(I) = XORD(K)
  520 Y(I) = YORD(K)
      X(5) = XCENT(N)
      Y(5) = YCENT(N)
      THICK = TH(N)
      XU = PR(M)
      E = YM(M)
      CALL STCUAD
  550 CALL WRDISK (NTRACK,S11,256)
      NTRACK = NTRACK + 1
      CALL WRDISK (NTRACK,S21,260)
      NTRACK = NTRACK + 1
      GC TO 7CC
C
C     TRIANGLE
C
  600 CO 620  I = 1,3
      J = IPERM(I)
      L = IPERM(J)
      K1 = NP(N,I)
      K2 = NP(N,J)
      A(L)  = XCRC(K2)-XORD(K1)
  620 B(L) = YORD(K1)-YORD(K2)
      AREA = A(3)*B(2)-A(2)*B(3)
      ET = YM(M)
      NU = PR(M)
      THIK = TH(N)
      CALL STLST6
      CALL WRDISK (NTRACK,ST,256)
      NTRACK = NTRACK + 2
  700 CONTINUE
      NTRA = NTRACK + 1
  800 RETLRN
C
C     ****************************************************************
C     ERROR EXITS
```

```
C     *******************************************************************
C
 1000 PRINT 1001
 1001 FORMAT  (30HOMAX. NO. OF ELEMENTS EXCEEDED)
      IFLAG = 1
      GO TO 260
 1010 PRINT 1011
 1011 FORMAT (34HOMAX. NO. OF NODAL POINTS EXCEEDED)
      IFLAG = 1
      GO TO 270
 1020 PRINT 1021
 1021 FORMAT (33HOMAX. NO. OF CONSTRAINTS EXCEEDED)
      IFLAG = 1
      GO TO 280
 1040 PRINT 1041, K1
 1041 FORMAT  (28HOMISSING COORDINATE, POINT = I5)
      IFLAG = 1
      GO TO 400
 1050 PRINT 1051, N
 1051 FORMAT (34HONEGATIVE TRIANGLE AREA, ELEMENT = I5)
      IFLAG = 1
      GO TO 400
 1060 PRINT 1061
 1061 FORMAT(1HO,T3,'ERROR DETECTED IN READING INITIAL ''START'' CARD')
      STOP
      END
```

```
      SUBROUTINE STQUAD
C
C
C     ******************************************************************
C     THIS SUBROUTINE ASSEMBLES AND CCNDENSES THE STIFFNESS MATRIX OF
C     A CUADRILATERAL FCRMED BY FCUR LST ELEMENTS
C     ******************************************************************
C
      COMMON /QUACRG/ S11(16,16),S21(10,26), X(5), Y(5), E, XU, THICK
      COMMON /STFARG/ ST(16,16),B(3),A(3),AREA, ET,NU, THIK
      REAL NU
      DIMENSION S(26,26), LOC(12,4), IPERM4(4), DUM(1)
      EQUIVALENCE (CUM,S)
      CATA IPERM4/2,3,4,1/
      DATA LOC / 1, 2, 3, 4,17,18,  9,10,21,22,19,20,
     1           3, 4, 5, 6,17,18, 11,12,23,24,21,22,
     2           5, 6, 7, 8,17,18, 13,14,25,26,23,24,
     3           7, 8, 1, 2,17,18, 15,16,19,20,25,26 /
C
      ET = E
      THIK = THICK
      NU = XU
      DC 100 I = 1,676
  100 DUM(I) = 0.
C
C     ASSEMBLY OF FOUR SUBTRIANGLE STIFFNESSES
C
      DC 160 N = 1,4
      M = IPERM4(N)
      A(1) = X(5)-X(M)
      A(2) = X(N)-X(5)
      A(3) = X(M)-X(N)
      B(1) = Y(M)-Y(5)
      B(2) = Y(5)-Y(N)
      B(3) = Y(N)-Y(M)
      AREA = A(3)*B(2)-A(2)*B(3)
      CALL STLST6
      DO 160 I = 1,12
      K = LOC(I,N)
      DO 160 J = 1,12
      L = LOC(J,N)
  160 S(K,L) = S(K,L) + ST(I,J)
C
C     CONDENSATICN OF INTERNAL NODAL POINTS
C
      DO 200 N = 1,10
      K = 26 - N
```

```
      L = K + 1
      PIVOT = S(L,L)
      DO 200 J = 1,K
      C = S(L,J)/PIVOT
      S(L,J) = C
      DO 200 I = J,K
      S(I,J) = S(I,J) -C*S(I,L)
  200 S(J,I) = S(I,J)
      DO 220 I = 1,16
      DO 220 J = 1,16
  220 S11(I,J) = S(I,J)
      DO 250 I = 1,10
      DO 250 J = 1,26
  250 S21(I,J) = S(I+16,J)
      RETURN
      END
```

```
      SUBROUTINE STLST6
C
C
C     ****************************************************************
C     ELEMENT STIFFNESS SUBROUTINE
C     LINEARLY VARYING STRAIN TRIANGLE WITH SIX NODAL POINTS
C     LINEAR ELASTIC ISOTROPIC MATERIAL
C     ****************************************************************
C
      COMMON /STFARG/ ST(16,16), B(3), A(3), AREA, ET, NU, THIK
      REAL NU, NUH
      DIMENSION CX1(3), CX2(3), CX3(3), CY1(3), CY2(3), CY3(3),
     1 U(3,6), V(3,6), UV(3,6,2), BA(3,2), IPERM(3)
      EQUIVALENCE (BA,B),  (UV,U), (UV(19),V)
      DATA IPERM /2,3,1/
      NUH = 0.5*(1.-NU)
      ER = ET/(1.-NU*NU)
      COMM = ER*THIK/(24.*AREA)
      DO 150 L = 1,3
      L1 = IPERM(L)
      L2 = IPERM(L1)
      L3 = L + 3
      DO 150 N = 1,2
      D0 = BA(L,N)
      D1 = BA(L1,N)
      UV(L ,L,N) = 3.*D0
      UV(L1,L,N) =    -D0
      UV(L2,L,N) =    -D0
      UV(L ,L3,N) = 4.*D1
      UV(L1,L3,N) = 4.*D0
  150 UV(L2,L3,N) = 0.
      DO 300  I = 1,6
      DO 200  L = 1,3
      CX1(L) = (U(1,I)+U(2,I)+U(3,I)+U(L,I))*COMM
      CY2(L) = (V(1,I)+V(2,I)+V(3,I)+V(L,I))*COMM
      CX2(L) = CX1(L)*NU
      CY1(L) = CY2(L)*NU
      CX3(L) = CY2(L)*NUH
  200 CY3(L) = CX1(L)*NUH
      K2 = 2*I
      K1 = K2 - 1
      DO 300 J = I,6
      L2 = 2*J
      L1 = L2 - 1
      X1 = 0.
      X2 = 0.
      X3 = 0.
```

```
      X4 = 0.
      DO 280 K = 1,3
      X = U(K,J)
      Y = V(K,J)
      X1 = X1 + CX1(K)*X + CX3(K)*Y
      X2 = X2 + CX2(K)*Y + CX3(K)*X
      X3 = X3 + CY1(K)*X + CY3(K)*Y
  280 X4 = X4 + CY2(K)*Y + CY3(K)*X
      ST(K1,L1) = X1
      ST(L1,K1) = X1
      ST(K1,L2) = X2
      ST(L2,K1) = X2
      ST(K2,L1) = X3
      ST(L1,K2) = X3
      ST(K2,L2) = X4
  300 ST(L2,K2) = X4
      RETURN
      END
```

```
      SUBROUTINE LDINPT
C
C
C     ****************************************************************
C     LDINPT INPUTS LCAC CASE AND REDUCES THERMAL, GRAVITY AND
C     IN-PLANE DISTRIBUTED LOADS TO KINEMATICALLY EQUIVALENT
C     NODAL PCINT FCRCES
C     ****************************************************************
C
      CCMMON
     1 NUMEL, NUMCP, NUMNP, NUMBC, NLOAD, MAXIT, NEQ,
     2 IBANDW, NECBC, NRUN, NTRA, NSKEWB, LNQ(6,4),
     3 T1, T2, T3, T4, T5, THERL
      COMMON /CMATPR/  YM(6), PR(6), RHO(6), ALFA(6)
      COMMON /CELMAR/  NP(350,8), NEBC(250), BANGLE(250), DISPBC(250)
      COMMON /CNTARG/  NUMPB, NSK, PGRAPH, IGRTAG(6), SPACNG(6),
     1 GRHEAD(3,6), NPB(50), NELSKP(50)
      COMMON /CTHERM/ B(3), A(3), DLT(3), CCMM, FX(6), FY(6)
      DIMENSION  HEAD(20), CF(3,3), P(3,2), PC(3,2), PN(3), PT(3),
     1 NOD(3), DELT(4), IPERM(3), IPERM4(4), XLOAD(1050), YLOAD(1050),
     2 XORD(1050), YORD(1050), XCENT(350), YCENT(350), NEI(350),
     3 MAT(350), TH(350), DT(350,5), S21(10,16), S22(10,10), S23(26),
     4 ELOAD(350,26)
      EQUIVALENCE (PN,PC), (PT,PC(4))
      REAL*8  EFLAG, FLAG, CHECK
      LOGICAL THERL, DENS
      DATA  EFLAG /8HSTART   /,  FLAG /8HLCADING /
      CATA IPERM /2,3,1/, IPERM4 /2,3,4,1/
      DATA CF /4.,2.,-1., 2.,16.,2., -1.,2.,4./
C
  100 READ(5, 9)  CHECK
      IF  (CHECK.EQ.EFLAG)  GC TC 1000
      IF  (CHECK.NE.FLAG)  GC TO 100
    9 FORMAT (A8)
   10 FCRMAT  (20A4)
C
C     INITIALIZE
C
      REWIND 8
      REWIND 3
      READ (8)   (XCRD(N),N=1,NUMNP),(XCENT(N),N=1,NUMEL),
     1      (YORD(N),N=1,NUMNP),(YCENT(N),N=1,NUMEL)
      READ (8)   (MAT(N), N=1,NUMEL)
      READ (8)   (TH(N),  N=1,NUMEL)
      THERL = .FALSE.
      DO 110  N = 1,50
  110 NELSKP(N) = 0
```

```
      CC 120  N = 1,NUMNP
      XLCAD(N) = O.
  120 YLOAD(N) = C.
      DU 140  N = 1,NUMEL
      NEI(N) = 0
      DC 130  I = 1,5
  130 DT(N,I) = 0.
      CO 140  I = 1,26
  140 ELCAD(N,I) = C.
C
C     CONTROL CARC AND GRAPH SPECIFICATIONS
C
      READ(5, 1C)  HEAD
      READ(5, 2C)  NPLD, NELD, NTLD, DENS
   20 FCRMAT (3I4,L4)
      PRINT 25, NRUN, HEAD, NPLD, NELD, NTLD, DENS
   25 FCRMAT  (14H1LOAD CASE NO.  I5 // 1X, 20A4 //
     1 35H NO. CF NODAL PCINT LOAC CARDS . .  I8 /
     2 35H NO. OF ELEMENT LOAD CARDS . . . .   I8 /
     3 35H NO. OF THERMALLY LCADED ELEMENTS    I8 /
     4 35H FLAG FCR GRAVITY LOADING  . . . .   L8)
      READ(5, 28)  IGRTAG, NSK
   28 FORMAT (20I4)
      PRINT 30, (GRHEAD(1,N),GRHEAD(2,N),GRHEAD(3,N),IGRTAG(N),N=1,6)
   30 FORMAT (7H- GRAPH, 13X, 3HTAG/ 13X, 18H(1=PLOT,0=NO PLOT)  /
     1 (1X, 3A4, I9))
      IF (NSK.LE.C)  GO TC 150
      READ(5, 28)  (NELSKP(I), I=1,NSK)
      PRINT 31, (NELSKP(I),I=1,NSK)
   31 FORMAT (20HCSKIPPED ELEMENTS = 25I4 /(20X,25I4))
C
C     ****************************************************************
C     CONCENTRATED NODAL POINT FCRCES
C     ****************************************************************
C
  15C IF (NPLD.LE.C)  GO TO 200
      PRINT 33
   33 FCRMAT (26H-CONCENTRATED NOCAL FORCES //
     1 6H POINT, 6X, 6HX-LOAD, 8X, 6HY-LOAD /1X)
      DO 180  L = 1,NPLC
      READ(5, 35)  N, XLOAD(N), YLOAD(N)
  180 PRINT 38, N, XLCAD(N), YLOAD(N)
   35 FORMAT (I4, 2F8.3)
   38 FORMAT (I6, 2F14.5)
C
C     ****************************************************************
C     ELEMENT SIDE LOADING (NORMAL PRESSURE AND SURFACE SHEAR)
```

```
C     ********************************************************************
C
  200 IF (NELD.LE.0)  GO TO 300
      PRINT 40
   40 FORMAT (20H-ELEMENT SIDE FORCES  // 8H ELEMENT, 3X, 4HSIDE,
     1 4X, 4HNODE, 3X, 11HN. PRESSURE, 3X, 11HSURF. SHEAR)
      DO 280  L = 1,NELD
      READ(5, 45)  N, I, P
   45 FORMAT (2I4, 6F8.3)
      K1 = NP(N,I)
      IF (NP(N,7).LE.0)  GO TO 220
      J = IPERM4(I)
      M = NP(N,I+4)
      GO TO 230
  220 J = IPERM(I)
      M = NP(N,I+3)
  230 K2 = NP(N,J)
      NOD(1) = K1
      NOD(2) = M
      NOD(3) = K2
      PRINT 50, N, I, (NOD(J),P(J,1),P(J,2),J=1,3)
   50 FORMAT (1H0,2I7,I8,2F14.4/(15X,I8,2F14.4))
      X = XORD(K1)-XORD(K2)
      Y = YORD(K2)-YORD(K1)
      DO 250  K = 1,2
      DO 250  I = 1,3
      PC(I,K) = 0.
      DO 240  J = 1,3
  240 PC(I,K) = PC(I,K) + CF(I,J)*P(J,K)
  250 PC(I,K) = PC(I,K)*TH(N)/30.
      DO 270  I = 1,3
      K = NOD(I)
      XLOAD(K) = XLOAD(K) + PN(I)*Y - PT(I)*X
  270 YLOAD(K) = YLOAD(K) + PN(I)*X + PT(I)*Y
  280 CONTINUE
C
C     ********************************************************************
C     GRAVITY LOADS
C     ********************************************************************
C
  300 IF (.NOT.DENS)  GO TO 400
      DO 380  N = 1,NUMEL
      M = MAT(N)
      IF (NP(N,7).LE.0)  GO TO 360
      NEI(N) = 1
      DO 350  I = 1,4
      J = IPERM4(I)
```

```
      K1 = NP(N,I)
      K2 = NP(N,J)
      A3 = XORD(K2)-XORD(K1)
      B3 = YORD(K1)-YORD(K2)
      A2 = XORD(K1)-XCENT(N)
      B2 = YCENT(N)-YORD(K1)
      AREA = A3*B2-A2*B3
      COMM = RHO(M)*AREA/6.
      DO 350  L = 4,6
      K = 2*LNQ(L,I)
  350 ELOAD(N,K) = ELOAD(N,K) -COMM
      GO TO 380
  360 K1 = NP(N,1)
      K2 = NP(N,2)
      K3 = NP(N,3)
      A3 = XORD(K2)-XORD(K1)
      B3 = YORD(K1)-YORD(K2)
      A2 = XORD(K1)-XORD(K3)
      B2 = YORD(K3)-YORD(K1)
      AREA = A3*B2 - A2*B3
      COMM = RHO(M)*AREA/6.
      DO 370  L = 4,6
      K = NP(N,L)
  370 YLOAD(K) = YLOAD(K) - COMM
  380 CONTINUE
C
C     ******************************************************************
C     THERMAL LOADS
C     ******************************************************************
C
  400 IF (NTLD.LE.0)  GO TO 500
      PRINT 60
   60 FORMAT (19H-THERMAL INCREMENTS // 8H ELEMENT, 6X, 8HCORNER I,
     1 6X, 8HCORNER J, 6X, 8HCORNER K, 6X, 8HCORNER L /1X)
      DO 480  L = 1,NTLD
      THERL = .TRUE.
      READ(5, 65)  N, DELT
      PRINT 68, N, DELT
   65 FORMAT (I4,4F10.3)
   68 FORMAT (I8,4F14.4)
      M = MAT(N)
      COMM = ALFA(M)*YM(M)*TH(N)/(6.*(1.-PR(M)))
      DO 420  I = 1,4
  420 DT(N,I) = DELT(I)
      IF  (NP(N,7).LE.0)  GO TO 450
      NEI(N) = 1
      AVDT = (DELT(1)+DELT(2)+DELT(3)+DELT(4))/4.
```

```
      DT(N,5) = AVDT
      DLT(3) = AVCT
      DO 440  I = 1,4
      J = IPERM4(I)
      K1 = NP(N,I)
      K2 = NP(N,J)
      A(1) = XCENT(N)-XCRD(K2)
      A(2) = XORD(K1)-XCENT(N)
      A(3) = XORD(K2)-XCRD(K1)
      B(1) = YORD(K2)-YCENT(N)
      B(2) = YCENT(N)-YCRD(K1)
      B(3) = YORD(K1)-YCRD(K2)
      DLT(1) = DELT(I)
      DLT(2) = DELT(J)
      CALL THERLD
      DO 440  J = 1,6
      K = 2*LNQ(J,I)
      ELOAD(N,K-1) = ELCAC(N,K-1) + FX(J)
  440 ELOAD(N,K)   = ELCAD(N,K)   + FY(J)
      GC TC 480
  450 DC 460  I = 1,3
      J = IPERM(I)
      K1 = NP(N,I)
      K2 = NP(N,J)
      M = IPERM(J)
      A(M) = XORD(K2)-XORD(K1)
      B(M) = YORD(K1)-YORD(K2)
  46C DLT(I) = DELT(I)
      CALL THERLD
      DC 47C  J = 1,6
      K = NP(N,J)
      XLOAD(K) = XLOAC(K) + FX(J)
  47C YLOAD(K) = YLOAD(K) + FY(J)
  480 CONTINUE
      WRITE (3)  ((CT(N,I),I=1,5),N=1,NUMEL)
C
C     ********************************************************************
C     CONDENSATION CF INTERNAL QUADRILATERAL LOADS
C     ********************************************************************
C
  500 DC 6CC N = 1,NUMEL
      IF (NEI(N).LE.0)  GC TO 600
      NTRACK = 2*N - 1
      CALL RDDISK (NTRACK,S21,260)
      DO 510  I = 1,26
  51C S23(I) = ELOAD(N,I)
      DO 530  II = 1,10
```

```
      K = 26 - II
      L = K  + 1
      M = L - 16
      DO 520  I = 1,K
  520 S23(I) = S23(I) - S21(M,I)*S23(L)
  530 S23(L) = S23(L)/S22(M,M)
      CALL WRDISK (NTRACK,S21,286)
C
C     TRANSFER TC EXTERNAL NODE FCRCE VECTOR
C
      DO 540  I = 1,8
      J = 2*I
      K = NP(N,I)
      XLOAD(K) = XLOAD(K) + S23(J-1)
  540 YLOAD(K) = YLOAD(K) + S23(J)
  600 CONTINUE
      WRITE (3) (NEI(N),N=1,NUMEL)
C
C     *******************************************************************
C     IMPOSE PARTIAL FORCE VECTOR B.C. AND STORE ON TAPE
C     *******************************************************************
C
      REWIND 2
      DC 700  M = 1,NEQBC
      N = NEBC(M)
      L = (N+1)/2
      PHI = BANGLE(M)
      IF (PHI.NE.C.)  XLOAD(L) = XLOAD(L)*COS(PHI) + YLOAD(L)*SIN(PHI)
      K = 2*L - N
      IF(DISPBC(L).NE.0.0) GC TC 700
      IF (K.EQ.1)  XLOAD(L) = 0.
      IF (K.EQ.0)  YLOAD(L) = 0.
  700 CONTINUE
      WRITE (2)  (XLOAD(N),YLCAD(N), N=1,NUMNP)
      RETURN
C
C     ERROR EXIT
C
 1000 PRINT 1001
 1001 FORMAT (46HCSTART CARD FOUND WHEN LOOKING FOR LOADNG CARD)
      STOP
      END
```

```
      SLBRCLTINE THERLD
C
C
C     **********************************************************************
C     THIS SUBROUTINE CCMPUTES RESTRAINT THERMAL FORCES FOR
C     A SIX NODAL PCINT TRIANGLE
C     **********************************************************************
C
      CCMMCN /CTHERM/ B(3), A(3), DLT(3), COMM, FX(6), FY(6)
      DIMENSICN IPERM(3)
      CATA IPERM /2,3,1/
      DO 120  I = 1,12
  120 FX(I) = 0.
      DC 150  I = 1,3
      J = IPERM(I)
      K = IPERM(J)
      DT3 = DLT(K)
      DT5 = DLT(I) + 2.*DLT(J) + DT3
      DT6 = DLT(J) + 2.*DLT(I) + DT3
      X = B(K)*CCMM
      Y = A(K)*CCMM
      FX(K) = X*DT3
      FY(K) = Y*DT3
      FX(J+3) = FX(J+3) + X*DT5
      FY(J+3) = FY(J+3) + Y*DT5
      FX(K+3) = FX(K+3) + X*DT6
  15C FY(K+3) = FY(K+3) + Y*DT6
      RETLRN
      END
```

```
      SUBROUTINE FORMK
C
C
C     ******************************************************************
C     FORMK ASSEMBLES THE CCMPLETE STIFFNESS MATRIX
C     AND IMPCSES BCUNDARY CCNDITIONS.
C     ******************************************************************
C
      COMMCN
     1 NUMEL, NUMCP, NUMNP, NUMBC, NLCAD, MAXIT, NN, MM, NEQBC
      COMMON /CELMAR/  NP(350,8), NEBC(250), BANGLE(250), DISPBC(250)
      DIMENSION  NE(150,2C), NEB(20), IND(20), ST(16,16), NEL(150),
     1 S(17600), R(21C0)
      EQUIVALENCE (NE,NEL)
C
C
C     ******************************************************************
C     FINC ALL ELEMENTS CCNTRIBUTING TO EACH BLOCK OF 'NPBL'
C     NODAL PCINTS
C     ******************************************************************
C
      REWIND 2
      READ (2) (R(N), N=1,NN)
      NDIMS = 176C0
      NEQBL = NDIMS/MM
      NPBL = NEQBL/2
      MM1 = MM + 1
      MM2 = MM + MM
      NB = 1 + (NUMNP-1)/NPBL
      DO 120 I = 1,NB
      NEB(I) = 0
      DO 120 N = 1,15C
  120 NE(N,I) = 0
      CO 14C N = 1,NUMEL
      DO 130 I = 1,NB
  13C IND(I) = C
      DC 140 I = 1,8
      K = NP(N,I)
      IF (K.LE.C) GC TO 14C
      M = 1 + (K-1)/NPBL
      IF (IND(M).NE.O) GO TC 140
      NEB(M) = NEB(M) + 1
      L = NEB(M)
      NE(L,M) = N
      IND(M) = 1
  14C CONTINUE
      IF (NB.LE.1) GC TC 160
```

```
      REWIND 9
      DO 150 M = 2,NB
      L = NEB(M)
  150 WRITE (9) L, (NE(I,M), I=1,L)
      REWIND 9
C
C     ****************************************************************
C     GENERATION OF COMPLETE STIFFNESS MATRIX
C     ****************************************************************
C
  160 LL = NEB(1)
      REWIND 1
      DO 400 M = 1,NB
C
C     ASSEMBLE 'M-TH' BLOCK
C
      IF (M.GT.1) READ (9) LL, (NEL(I),I=1,LL)
      N1 = 1 + (M-1)*NPBL
      N2 = MIN0 (N1+NPBL-1,NUMNP)
      NPTB = N2 - N1 + 1
      NEQB = 2*NPTB
      NE1 = 2*N1 - 1
      NE2 = 2*N2
      N = MM*NEQB
      DO 180 I = 1,N
  180 S(I) = 0.
      DO 220 NV = 1,LL
      N = NEL(NV)
      NTRACK = 2*(N-1)
      CALL RDDISK (NTRACK,ST,256)
      DO 220 J = 1,8
      L = NP(N,J)
      IF (L.LT.N1.OR.L.GT.N2)  GO TO 220
      NC = MM2*(L-N1) + 1
      JJ = J + J - 1
      DO 200 I = 1,8
      K = NP(N,I)
      IF (K.LT.L.OR.K.GT.L) GO TO 200
      L1 = NC + 2*(L-K)
      L2 = L1 + MM
      II = I + I - 1
      S(L1)   = S(L1)   + ST(II  ,JJ  )
      S(L2)   = S(L2)   + ST(II+1,JJ+1)
      S(L2+1) = S(L2+1) + ST(II  ,JJ+1)
      IF (K.EQ.L) GO TO 200
      S(L1-1) = S(L1-1) + ST(II+1,JJ  )
  200 CONTINUE
```

```
  220 CONTINUE
C
C     IMPOSE BOUNDARY CONDITIONS
C
      DO 360 I = 1,NEQBC
      DISP = DISPBC(I)
      N = NEBC(I)
      NC = N - NE1 + 1
      IF (NC.LE.0.OR.NC.GT.NEQB) GO TO 300
      L = MM*(NC-1) + 1
      PHI = BANGLE(I)
      IF (PHI.EQ.0.) GO TO 270
      CN = COS(PHI)
      SN = SIN(PHI)
      L1 = L - MM
      S(L1) = S(L1)*CN*CN + 2.*S(L+1 )*SN*CN + S(L)*SN*SN
      L2 = L + 1
      DO 260 J = 3,MM
      L1 = L1 + 1
      L2 = L2 + 1
  260 S(L1) = S(L1)*CN + S(L2)*SN
  270 S(L) = 1.
      NDPL = N
      R(N) = DISP
      DO 280 J = 2,MM
      L = L + 1
      NDPL = NDPL - 1
      IF(NDPL.LE.0.OR.DISP.EQ.0.0) GO TO 280
      R(NDPL) = R(NDPL) - S(L)*DISP
  280 S(L) = 0.
  300 NCMIN = MAX0 (NC+1,1)
      NCMAX = MIN0 (NC+MM-1,NEQB)
      IF (NCMAX.LT.NCMIN) GO TO 360
      LL = MM1*NCMIN - MM2 - NC
      IF (PHI.EQ.0.) GO TO 340
      L = LL + 1
      K = NCMAX - 1
      IF (NCMAX.EQ.NEQB) K = NCMAX
      DO 330 J = NCMIN,K
      L = L + MM1
  330 S(L) = S(L)*CN + S(L-1)*SN
  340 L = LL
      NDPL = NCMIN + NE1 - 2
      DO 350 J = NCMIN,NCMAX
      NDPL = NDPL + 1
      L = L + MM1
      IF(NDPL.GT.NN.OR.DISP.EQ.0.0) GO TO 350
```

```
      R(NDPL) = R(NDPL) - S(L)*DISP
  350 S(L) = 0.
  360 CONTINUE
C
C     WRITE BLOCK OF EQUATIONS ONTO TAPE
C
      L1 = 1
      L2 = MM
      DO 380 N = 1,NEQB
      WRITE (1) (S(I),I=L1,L2)
      L1 = L1 + MM
  380 L2 = L2 + MM
  400 CONTINUE
      NLINE = (NUMNP-1)/2 + 1
      NPAGE = (NLINE-1)/70 + 1
      DO 420 IPG = 1,NPAGE
      WRITE(6,10)
   10 FORMAT('1NODAL FORCE VECTOR',//,2(7H  POINT,8X,6HX-LOAD,8X,
     1 6HY-LOAD,12X),//)
      DO 410 JJ=1,70
      I = 70*(IPG-1) + JJ
      IF(I.GT.NLINE) GO TO 420
      I1 = 2*(I-1) + 1
      I2 = I1 + 1
      IR1 = 2*I1 - 1
      IR2 = IR1 + 1
      IR3 = 2*I2 - 1
      IR4 = IR3 + 1
  410 WRITE(6,20) I1,R(IR1),R(IR2),I2,R(IR3),R(IR4)
  420 CONTINUE
   20 FORMAT(I6,6X,1P2E14.5,I12,6X,1P2E14.5)
      REWIND 2
      WRITE(2) (R(I), I=1,NN)
      RETURN
      END
```

```
      SUBROUTINE SOLVE
C
C
C     ******************************************************************
C     SOLVE OBTAINS NODAL POINT DESPLACEMENTS FROM THE LARGE
C     CAPACITY BAND SCLVER BIGSOL
C     ******************************************************************
C
      COMMON
     1 NUMEL, NUMCP, NUMNP, NUMBC,NLOAD, MAXIT, NEQ,
     2 IBANDW, NECBC, NRUN, NTRA, NSKEWB
      COMMON /CELMAR/  NP(350,8), NEBC(250), BANGLE(250), DISPBC(250)
      COMMON /BANARG/ NN, MM, MAXIR, TOLER, NTR, NITER, WS(15500)
      DIMENSION R(2100)
      EQUIVALENCE (WS,R)
      MM = IBANDW
      NN = NEQ
      MAXIR = MAXIT
      TOLER = 0.CC1
      NTR = NTRA
      KKK = 0
      IF (NRUN.GT.1)  KKK = 2
      CALL BIGSCL (KKK)
C
C     TRANSFORM SKEW CISPLACEMENTS, IF ANY, TO THE X Y GLOBAL SYSTEM
C
      IF (NSKEWB.LE.C)  GC TC 200
      DO 150 M = 1,NEQBC
      N = NEBC(M)
      PHI = BANGLE(M)
      IF (PHI.EQ.C.) GO TC 150
      L = N - 1
      R(N) = R(L)*SIN(PHI)
      R(L) = R(L)*COS(PHI)
  150 CONTINUE
C
C     PUT DISPLACEMENTS ON TAPE
C
  200 REWIND 2
      WRITE (2) (R(I),I=1,NN)
      RETURN
      END
```

```
      SUBROUTINE BIGSOL(KKK)
C
C
C     ******************************************************************
C     LINEAR EQUATION SOLVER FOR LARGE SYMMETRIC BAND MATRICES
C     ******************************************************************
C     CARLOS A. FELIPPA, JULY 1966.
C
C     INPUT
C
C     NN    = NUMBER OF EQUATIONS.
C     MM    = HALF BAND WIDTH.
C     MAXIT = MAX. NO. OF ITERATIONS ON RESIDUALS.
C     TOLER = ACCURACY TEST (VALID ONLY IF MAXIT GT 0).
C     NTR   = NUMBER OF DISK TRACK STARTING AT WHICH THE REDUCED MATRIX
C             IS STORED - TRACKS 'NTR' TO 'NTR+NREC', WHERE NREC =
C             (NN+NC-1)*NBUFF/NC, NC = 460*NBUFF/MM AND NBUFF = 1 IN
C             THIS VERSION, ARE USED FOR SUCH PURPOSE - WHEN RESIDUAL
C             ITERATIONS ARE PERFORMED, THE NEXT CYLINDER IS USED
C             TO STORE SUCCESIVE ITERATES OF THE SOLUTION VECTOR.
C     UPPER HALF BAND OF INPUT MATRIX IS READ COLUMN-WISE FROM
C     LOGICAL UNIT 1, ONE COLUMN PER RECORD.
C     INPUT VECTOR IS READ FROM LOGICAL UNIT 2.
C
C     STORAGE
C     WS    = WORKING SPACE OF LENGTH 'NWS' (SEE WRITE-UP) - WS CONTAINS
C       A  = STORAGE OF AN UPPER TRIANGLE OF BAND DURING REDUCTION.
C       D  =  DOUBLE PRECISION VECTOR FOR ITERATION ON RESIDUALS.
C       R1 =  SINGLE PRECISION VECTOR, EQUIVALENT TO D.
C       F  =  STORAGE OF FIST ROW DURING REDUCTION.
C       ND =  INDEXING ARRAY FOR THE UPPER TRIANGLE IN A.
C       X  =  BUFFER STORAGE FOR DISK I/O.
C
C     OUTPUT
C
C     R     = SOLUTION VECTOR, STORED IN THE FIRST NN LOCATIONS OF WS.
C     NITER = RETURNS NUMBER OF RESIDUAL ITERATIONS PERFORMED.
C
C     ARGUMENT KKK  SPECIFIES THE FOLLOWING OPTIONS
C        KKK = 0   MATRIX REDUCTION AND SUBSTITUTION OF INPUT VECTOR.
C        KKK = 1   MATRIX REDUCTION ONLY.
C        KKK = 2   SUBSTITUTION OF INPUT VECTOR ONLY.
C
C     THE LENGTH OF WS HERE CORRESPONDS TO A MAX. BAND WIDTH MM = 160.
C
      COMMON /BANARG/ NN, MM, MAXIT, TOLER, NTR, NITER, WS(15500)
      DIMENSION A(1), F(1), ND(1), R(1),R1(1),X(2300)
```

```
      DOUBLE PRECISICN C(1), D1, C2, C3
      EQUIVALENCE (WS,A,R), (WS(4401),D,R1), (WS(12881),F),
     1 (WS(13041),NC), (WS(13201),X)
      LOGICAL T1
C
      NR = NN - 1
      NM = NN - MM
      NN1 = NN + 1
      MM1 = MM + 1
      NITER = 0
      T1 = .FALSE.
      NBUFF = 1
      NC = (460*NBUFF)/MM
      NW = NC*MM
      NREC1 = (NN+NC-2)/NC -1
      IF (KKK.GE.2)  GC TC 210
C
C     ****************************************************************
C     DECOMPOSITICN OF BAND MATRIX
C     ****************************************************************
C
      DC 110  J = 1,MM
  110 ND(J) = (J*(J+1))/2
C
C     SET UP FIRST TRIANGULAR BLCCK IN A
C
      REWIND 1
      DO 130  N= 1,MM
      LC1 = ND(N) - N + 1
      LC2 = LC1 + MM - 1
  130 READ (1) (A(I),I=LC1,LC2)
      NX = 0
      NTRACK = NTR
      DO 200  N = 1,NR
C
C     TRIANGLE IS SIMULTANECUSLY REDUCED AND SHIFTED
C     PIVOTS AND MULTIPLIERS ARE TRANSFERRED TO X
C
      MR = MINO (MM,NN1-N)
      JJ = NX*MM + 1
      NX = NX + 1
      PIVCT = A(1)
      X(JJ) = PIVOT
      DO 150  J = 2,MR
      L = ND(J)
  150 F(J) = A(L)
      DO 160  J = 2,MR
```

```
      C = F(J)/PIVOT
      JJ = JJ + 1
      X(JJ) = C
      L  = ND(J)
      L1 = ND(J-1) + 1
      DO 160  I = 2,J
      L = L - 1
      L1 = L1 - 1
  16C A(L1) = A(L) - C*F(I)
      IF (N.GT.NM)  GC TO 190
C
C     STORE NEXT CCLUMN
C
      READ(1) (A(I),I=LC1,LC2)
  19C IF (NX.LT.NC)  GO TC 200
C
C     'NC' RECUCED RCWS ARE WRITTEN ON 'NBUFF' DISK TRACKS
C
      CALL WRDISK (NTRACK,X,NW)
      NTRACK = NTRACK + NBUFF
      NX = 0
  200 CONTINUE
      JJ = NX*MM + 1
      X(JJ) = A(1)
      CALL WRDISK (NTRACK,X,JJ)
      IF (KKK.EQ.1)  RETURN
C
C     ******************************************************************
C     SUBSTITUTICN CF INPUT VECTCR
C     ******************************************************************
C
  21C REWIND 2
      READ (2)  (R(I), I=1,NN)
C
C     FORWARD REDLCTICN
C
  22C NTRACK = NTR
      NX = NC
      DC 240  N = 1,NR
      MR = MINO (MM,NN1-N)
      IF (NX.LT.NC)  GC TC 230
      CALL RDDISK (NTRACK,X,NW)
      NTRACK = NTRACK + NBUFF
      NX = 0
  23C JJ = NX*MM + 1
      NX = NX + 1
      C = R(N)
```

```
      R(N) = C/X(JJ)
      I1 = N + 1
      I2 = I1 + MR - 2
      DO 240  I = I1,I2
      JJ = JJ + 1
  240 R(I) =R(I) - C*X(JJ)
      JJ = NX*MM + 1
      ALAST = X(JJ)
      IF (NX.GE.NC)  CALL RDDISK (NTRACK,ALAST,1)
      R(NN) = R(NN)/ALAST
      NTRES = NTRACK + 1
C
C     BACK SUBSTITUTICN
C
      NTRACK = NTRACK - NBUFF
      CALL RDDISK (NTRACK,X,NW)
      NX = NN - NREC1*NC - 1
      DO 260  L = 2,NN
      N = NN1 - L
      MR = MINO (MM,L)
      NX = NX - 1
      IF (NX.GE.0)  GC TO 250
      NTRACK = NTRACK - NBUFF
      CALL RDDISK (NTRACK,X,NW)
      NX = NC - 1
  250 JJ = NX*MM + 1
      I1 = N + 1
      I2 = I1 + MR - 2
      DO 260  I = I1,I2
      JJ = JJ + 1
  260 R(N) = R(N) -X(JJ)*R(I)
      IF (T1)  GC TC 400
      IF (MAXIT.LE.C)  RETURN
C
C     ***************************************************************
C     ITERATION ON RESIDUALS
C     ***************************************************************
C
  280 NITER = NITER + 1
      CALL WRDISK (NTRES,R,NN)
      REWIND 1
      READ (1)  (X(I),I=1,MM)
      D1 = X(1)
      D3 = R(1)
      D(1) = D1*D3
      DO 350  N = 2,NN
      MR = MINO(N,MM)
```

```
      READ (1)  (X(I),I=1,MM)
      D1 = X(1)
      C3 = R(N)
      C(N) = D1*D3
      K = N
      DO 350  J = 2,MR
      K = K - 1
      D1 = X(J)
      D2 = R(K)
      D(K) = D(K) + D1*D3
  350 D(N) = D(N) + D1*D2
      REWIND 2
      READ (2)  (R(I), I=1,NN)
      DO 360  N = 1,NN
      D1 = R(N)
  360 R(N) = D1 - D(N)
      T1 = .TRUE.
      GO TO 220
  400 CALL RDDISK (NTRES,R1,NN)
C
C     CHECK ACCURACY OF SCLUTION
C
      RNORM = 0.
      DELR = 0.
      DO 450  N = 1,NN
      RNCRM = RNORM + R1(N)**2
      DELR  = DELR  + R (N)**2
  450 R(N) = R1(N) + R(N)
      EPS = SQRT (DELR/RNCRM)
      IF (EPS.LE.TCLER) RETURN
      IF (NITER.LT.MAXIT) GO TO 280
      PRINT 99, NITER, EPS
   99 FORMAT (35HCSFECIFIED ACCURACY NOT ATTAINED IN I5, 24H ITERATIONS,
     1 LAST EPS = E14.4)
      RETURN
      END
```

```
      SUBROUTINE STRESS
C
C
C     ******************************************************************
C     STRESS COMPUTES AND PRINTS ELEMENT AND NODAL POINT STRESSES
C     ******************************************************************
C
      COMMON
     1 NUMEL, NUMCP, NUMNP, NUMBC, NLOAD, MAXIT, NEQ,
     2 IBANDW, NEQBC, NRUN, NTRA, NSKEWB, LNQ(6,4),
     3 T1, T2, T3, T4, T5, THERL
      COMMON /CELMAR/  NP(350,8), NEBC(250), BANGLE(250), DISPBC(250)
      COMMON /CMATPR/  YM(6), PR(6), RHO(6), ALFA(6)
      COMMON /CTRIST/ ER, G, NU, CODIL, DELT(3), B(3), A(3),
     1 RX(6), RY(6), ESIG(6,3)
      COMMON /CNTARG/  NUMPB, NSK, PGRAPH, IGRTAG(6), SPACNG(6),
     2 GRHEAD(3,6), NPB(50), NELSKP(50)
      DIMENSION  DSX(1050), DSY(1050), ANGLE(1050), ANGLEC(350),
     1 SIG(1050,7), SIGC(350,7), SIGM(1050,3), COUNT(1050),
     2 XCRD(1050), YCRD(1050), XCENT(350), YCENT(350),
     3 MAT(350), NEI(350), DT(350,3), D(26), SIGQ(9,3), COEF(4),
     4 FMAX(10), IPERM(3), IPERM4(4), S21(10,26), S23(26)
      EQUIVALENCE  (DSX,SIG(3151)), (DSY,DSX(1051)), (DT,DSY(1051)),
     1 (MAT,SIGC(1051)), (NEI,MAT(351)), (ANGLE,SIG(6301)),
     2 (ANGLEC,SIGC(2101))
      REAL NU
      LOGICAL T1, T2, T3, T4, T5, THERL, PGRAPH
      DATA IPERM /2,3,1/, IPERM4 /2,3,4,1/
      DATA COEF /0.50,0.50,0.25,1.00/
      DATA FMAX /10.,15.,20.,25.,30.,40.,50.,60.,80.,100./
C
C     ******************************************************************
C     PRINTOUT OF DISPLACEMENTS
C     ******************************************************************
C
      REWIND 2
      REWIND 3
      REWIND 8
      REWIND 9
      READ (2)  (DSX(I),DSY(I),I=1,NUMNP)
      PRINT 15
   15 FORMAT (26H1NODAL POINT DISPLACEMENTS  // 2(6H POINT, 7X,
     1 5HX-DIS, 9X, 5HY-DIS, 14X) /1X)
      PRINT 16, (N, DSX(N), DSY(N), N=1,NUMNP)
   16 FORMAT(5(I6,4X,1P2E14.5,I14,4X,1P2E14.5,/)/)
      READ (8)  (XCRD(N),N=1,NUMNP),(XCENT(N),N=1,NUMEL),
     1          (YCRD(N),N=1,NUMNP),(YCENT(N),N=1,NUMEL)
```

```
      READ (8)  (MAT(I),I=1,NUMEL)
      IF (THERL)  READ (3)  ((DT(N,I),I=1,5),N=1,NUMEL)
      READ (3)  (NEI(N),N=1,NUMEL)
C
C     PUNCH OF DISPLACEMENTS
C
      IF(T2) WRITE(4,3) (N,XORD(N),YORD(N),DSX(N),DSY(N), N=1,NUMNP)
    3 FORMAT (I4,2F8.3,2E14.5)
C
C     *******************************************************************
C     INITIALIZE FCR STRESS CCMPUTATICN
C     *******************************************************************
C
      DO 120  N = 1,NUMNP
      COUNT(N) = C.
      DO 12C  I = 1,3
      SIG (N,I) = 0.
  120 SIGM(N,I) = C.
      DC 130  N = 1,NUMEL
      DO 125  I = 1,5
  125 DT(N,I) = C.
      DO 130  I = 1,3
  13C SIGC (N,I) = C.
C
C     *******************************************************************
C     COMPUTATICN CF ELEMENT STRESSES
C     *******************************************************************
C
      IF (T4)  PRINT 20
   20 FORMAT (17H1ELEMENT STRESSES // 10H   ELEMENT,  13X,
     1 8HN' PCINT, 9X, 6HSIG-XX, 8X, 6HSIG-YY, 8X, 6HTAU-XY,//)
      DO 300 N = 1,NUMEL
      M = MAT(N)
      NU = PR(M)
      ER = YM(M)/(1.-NU**2)
      G = 0.5*ER*(1.-NU)
      CODIL = ALFA(M)
      IF (NP(N,7).LE.C)  GC TO 250
C
C     QUADRILATERAL ELEMENT
C
C     RECOVER DISPLACEMENTS CF INTERNAL PCINTS
C
      CC 150  I = 17,26
  15C S23(I) = C.
      CO 160  I = 1,8
      K = NP(N,I)
```

```
      L = 2*I
      D(L-1) = DSX(K)
  16C D(L)  =  DSY(K)
      NTRACK = 2*N - 1
      Nw = 26C
      IF (NEI(N).GT.0)  Nw = 286
      CALL RDDISK (NTRACK,S21,NW)
      DO 18C  I = 1,1C
      L = I + 16
      K = L - 1
      D(L) = S23(L)
      DO 180  J = 1,K
  18C D(L) = D(L) - S21(I,J)*D(J)
C
C     STRESSES ARE NOW EVALUATED AT EACH SUBTRIANGLE
C     AND AVERAGED FOR THE QUADRILATERAL
C
      DC 200  I = 1,27
  20C SIGQ(I,1) = C.
      DO 220  I = 1,4
      J = IPERM4(I)
      K1 = NP(N,I)
      K2 = NP(N,J)
      A(1) = XCENT(N)-XORD(K2)
      A(2) = XORD(K1)-XCENT(N)
      A(3) = XORD(K2)-XORD(K1)
      B(1) = YORD(K2)-YCENT(N)
      B(2) = YCENT(N)-YORD(K1)
      B(3) = YORD(K1)-YORD(K2)
      DO 210  L = 1,6
      K = 2*LNQ(L,I)
      RX(L) = D(K-1)
  210 RY(L) = D(K)
      DELT(1) = DT(N,I)
      DELT(2) = DT(N,J)
      DELT(3) = DT(N,5)
      CALL TRISTR
      DO 220  K = 1,4
      M = LNQ(K,I)
      DO 220  J = 1,3
  220 SIGQ(M,J) = SIGQ(M,J) + COEF(K)*ESIG(K,J)
      IF (T4)  PRINT 25, N, (NP(N,I),(SIGQ(I,J),J=1,3),I=1,4),
     1 (SIGQ(9,J),J=1,3)
   25 FORMAT (14H0QUADRILATERAL I4, 4X, 6HCORNER I4, 3F14.5,/
     13(22X,6HCORNER,I4, 3F14.5,/),22X,10HCENTROID  , 3F14.5)
      DO 230  I = 1,8
      K = NP(N,I)
```

```
      CCUNT (K) = CCUNT(K) + 1.
      DO 230  J = 1,3
      X = SIGQ(I,J)
      IF (ABS(X).GT.ABS(SIGM(K,J)))  SIGM(K,J) = X
  230 SIG(K,J) = SIG(K,J) + X
      DO 240  J = 1,3
  240 SIGC(N,J) = SIGC(S,J)
      GO TO 300
C
C     SINGLE TRIANGLE
C
  250 DO 260  I = 1,3
      J = IPERM(I)
      M = IPERM(J)
      CELT(I) = DT(N,I)
      K1 = NP(N,I)
      K2 = NP(N,J)
      A(M) = XORD(K2)-XCRD(K1)
  260 B(M) = YCRD(K1)-YCRD(K2)
      DO 270  I = 1,6
      K = NP(N,I)
      RX(I) = DSX(K)
  270 RY(I) = DSY(K)
      CALL TRISTR
      DO 280  I = 1,6
      K = NP(N,I)
      CCUNT(K) = CCUNT(K) + 1.
      DO 280  J = 1,3
      X = ESIG(I,J)
      IF  (ABS(X).GT.ABS(SIGM(K,J)))  SIGM(K,J) = X
  280 SIG(K,J) = SIG(K,J) + X
      IF (T4)  PRINT 26, N, (NP(N,I),(ESIG(I,J),J=1,3),I=1,3)
   26 FORMAT (9HOTRIANGLE I4, 9X, 6HCCRNER I4, 3F14.5 /
     1 (22X, 6HCCRNER I4, 3F14.5))
  300 CONTINUE
C
C     ***********************************************************************
C     NODAL PCINT AVERAGE STRESSES
C     ***********************************************************************
C
      DC 40C  N = 1,NUMNP
      DO 320  I = 1,3
  32C SIG(N,I) = SIG(N,I)/CCUNT(N)
      X = SIG(N,1)
      Y = SIG(N,2)
      XY= SIG(N,3)
      C = 0.5*(X+Y)
```

```
      DIF = 0.5*(X-Y)
      RR = SQRT(DIF**2 + XY**2)
      SIG(N,4) = C + RR
      SIG(N,5) = C - RR
      SIG(N,6) = RR
      ANGLE (N) = 45.
      IF (DIF.NE.0.)  ANGLE(N) = 28.647890*ATAN(XY/DIF)
  400 CONTINUE
      DO 420  N = 1,NUMEL
      IF (NP(N,7).LE.0)  GO TO 420
      X =  SIGC(N,1)
      Y =  SIGC(N,2)
      XY = SIGC(N,3)
      C = 0.5*(X+Y)
      DIF = 0.5*(X-Y)
      RR = SQRT(DIF**2 + XY**2)
      SIGC(N,4) = C + RR
      SIGC(N,5) = C - RR
      SIGC(N,6) = RR
      ANGLEC(N) = 45.
      IF (DIF.NE.0.)  ANGLEC(N) = 28.647890*ATAN(XY/DIF)
  420 CONTINUE
C
C     ******************************************************************
C     PRINT OF AVERAGED NODAL POINT STRESSES
C     ******************************************************************
C
      PRINT 30
      DO 440  N = 1,NUMNP
  440 PRINT 32,  N, XCRD(N),YORD(N),SIG(N,1),SIGM(N,1),SIG(N,2),
     1 SIGM(N,2),SIG(N,3),SIGM(N,3),(SIG(N,I),I=4,7)
      PRINT 36, NUMNP
      DO 450  N = 1,NUMEL
      IF (NP(N,7).LE.0)  GO TO 450
      L = N + NUMNP
      PRINT 38, L, XCENT(N),YCENT(N),(SIGC(N,I),I=1,7)
  450 CONTINUE
   30 FORMAT (30H1AVERAGED NODAL POINT STRESSES // 6H POINT,3X,
     1 11HCOORDINATES, 9X, 8HSIGMA-XX, 13X, 8HSIGMA-YY, 13X, 8H TAU-XY ,
     2 10X, 7HSIG-MAX, 4X, 7HSIG-MIN, 2X, 9HMAX.SHEAR, 5X, 5HANGLE /
     3 10X, 1HX, 7X, 1HY, 2X, 3(4X,7HAVERAGE,3X,7HMAXIMUM), 36X,
     4 11H(SIG-MAX,X) /1X)
   32 FORMAT (I5, 2F8.3, 3( F11.4, F10.4), 1X, 4F11.4)
   36 FORMAT (49H-STRESSES AT QUADRILATERAL CENTROIDS (POINT NO. = I5,
     1 15H + ELEMENT NO.)/1X)
   38 FORMAT (I5, 2F8.3, 3( F16.4, 5X), 1X, 4F11.4)
C
```

```
C     PUNCH OF STRESSES
C
      IF (.NOT.T3)  GO TO 500
      WRITE(4,5) (N, (SIG(N,I), I=1,3), N=1,NUMNP)
      DO 480  N = 1,NUMEL
      L = NUMNP + N
  480 WRITE(4,5) L, (SIGC(N,I), I=1,3)
    5 FORMAT (I4,3E18.6)
C
C     ******************************************************************
C     COMPUTATION OF GRAPH SPACINGS
C     ******************************************************************
C
  500 REWIND 9
      PGRAPH = .FALSE.
      DO 520  I = 1,6
      SPACNG(I) = 0.
      IF (IGRTAG(I).LE.0)  GO TO 520
      WRITE (9)  (SIG(N,I),N=1,NUMNP), (SIGC(N,I),N=1,NUMEL)
      PGRAPH = .TRUE.
  520 CONTINUE
      IF (.NOT.PGRAPH)  GO TO 800
      IF (NSK.LE.0)  GO TO 600
      DO 550  I = 1,NSK
      N = NELSKP(I)
      NEP = 8
      IF (NP(N,7).LE.0)  NEP = 6
      DO 550 J = 1,NEP
      K = NP(N,J)
      DO 550 L = 1,6
  550 SIG(K,L) = 0.
  600 DO 700  I = 1,6
      IF (IGRTAG(I).LE.0)  GO TO 700
      SGMAX = 0.
      DO 620  N = 1,NUMNP
      C = ABS(SIG(N,I))
  620 IF (C.GT.SGMAX)  SGMAX = C
      NF = ALOG10(SGMAX)
      N = 2
      IF (SGMAX.GE.1.)  N = 1
      F = 10.**(N-NF)
      C = F*SGMAX
      DO 630 L = 1,10
      IF (FMAX(L).GT.C)  GO TO 650
  630 CONTINUE
  650 SPACNG(I) = 0.1*FMAX(L)/F
  700 CONTINUE
```

```
C
  800 RETURN
      END
```

```
      SUBROUTINE TRISTR
C
C
C     ******************************************************************
C     TRISTR COMPUTES STRESSES FOR A 6 NODAL POINT TRIANGLE
C     ******************************************************************
C
      COMMON /CTRIST/ ER, G, NU, CODIL, DELT(3), B(3), A(3),
     1 RX(6), RY(6), ESIG(6,3)
      DIMENSION BA(3,2), UV(3,6,2), U(3,6), V(3,6), EPSX(3), EPSY(3),
     1 GMXY(3), IPERM(3)
      EQUIVALENCE (BA,B), (UV,U), (UV(19),V)
      REAL NU
      DATA IPERM /2,3,1/
C
      AREA = A(3)*B(2)-A(2)*B(3)
      DO 120  L = 1,3
      L1 = IPERM(L)
      L2 = IPERM(L1)
      L3 = L + 3
      DO 120  N = 1,2
      D0 = BA(L,N) /AREA
      D1 = BA(L1,N)/AREA
      UV(L ,L ,N) = 3.*D0
      UV(L1,L ,N) =   -D0
      UV(L2,L ,N) =   -D0
      UV(L ,L3,N) = 4.*D1
      UV(L1,L3,N) = 4.*D0
  120 UV(L2,L3,N) =    0.
      DO 150  I = 1,3
      THERM = CODIL*DELT(I)
      EPSX(I) = 0.
      EPSY(I) = 0.
      GMXY(I) = 0.
      DO 140  J = 1,6
      X = RX(J)
      Y = RY(J)
      C = U(I,J)
      D = V(I,J)
      EPSX(I) = EPSX(I) + C*X
      EPSY(I) = EPSY(I) + D*Y
  140 GMXY(I) = GMXY(I) + C*Y + D*X
      ESIG(I,1) = ER*(EPSX(I)+NU*EPSY(I)) - THERM
      ESIG(I,2) = ER*(EPSY(I)+NU*EPSX(I)) - THERM
  150 ESIG(I,3) = G* GMXY(I)
      DO 180  I = 1,3
      J = IPERM(I)
```

```
      DO 180  K = 1,3
  180 ESIG(I+3,K) = 0.5*(ESIG(I,K)+ESIG(J,K))
      RETURN
      END
```

```
      SUBROUTINE CNTPLT
C
C
C     ******************************************************************
C     CNTPLT PRINTS STRESS CONTOUR GRAPHS
C     ******************************************************************
C
      COMMON NUMEL, NUMCP, NUMNP
      COMMON /CELMAR/  NP(350,8)
      COMMON /CNTARG/  NUMPB, NSK, PGRAPH, IGRTAG(6), SPACNG(6),
     1 GRHEAD(3,6), NPB(50), NELSKP(50)
      DIMENSION XLAB(11), S(3), NR(2,3), NPT(3), IPERM4(4), P(101,101),
     1  XORD(1400), YORD(1400), F(1400)
      LOGICAL  T0, T1, T2, T3
      DATA  ASTRK /4H*   /,  BLANK /4H    /,
     1 XLAB(1) /44HC   1   2   3   4   5   6   7   8   9   D   /
      DATA IPERM4 /2,3,4,1/
      DATA NR /2,3,1,3,1,2/
C
      REWIND 8
      REWIND 9
      NTOTP = NUMNP + NUMEL
      READ (8)   (XORD(N),N=1,NTOTP),(YORD(N),N=1,NTOTP)
C
C     PREPARE GRAPH PARAMETERS
C
      I = NPB(1)
      XMIN = XORD(1)
      YMIN = YORD(1)
      YMAX = YMIN
      XMAX = XMIN
      DO 100  N = 2,NUMPB
      I = NPB(N)
      Y = YORD(I)
      X = XORD(I)
      IF (XMIN.GT.X)  XMIN = X
      IF (YMIN.GT.Y)  YMIN = Y
      IF (XMAX.LT.X)  XMAX = X
  100 IF (YMAX.LT.Y)  YMAX = Y
      XD = XMAX - XMIN
      YD = (YMAX-YMIN)*.6
      XM = XD
      IF (YD.GT.XD)  XM = YD
      DX = XM/100.
      XR = XMIN - DX
      DY = DX/.6
      YR = YMIN - DY
```

```
      XS = XR - DX/2.
      YS = YR - DY/2.
      NCOL1 = (YMAX - YS)/DY + 1.
      NRP = 101
      NCP = NCOL1-1
      NG = 0
C
C     CHECK IF GRAPH IS TO BE PLOTTED
C
  105 NG = NG + 1
      IF(NG.GT.6)  GO TO 600
      IF (IGRTAG(NG).LE.0)  GO TO 105
      READ (9)  (F(N),N=1,NTOTP)
      SPACE = SPACNG(NG)
      FMAX = 10.*SPACE
      DO 110  I = 1,NRP
      DO 110  J = 1,NCP
  110 P(I,J) = BLANK
C
C     PLOT BOUNDARY
C
      DO 160  N = 1,NUMPB
      K = NPB(N)
      L = NPB(N+1)
      IF (N.EQ.NUMPB)  L = NPB(1)
      X1 = XORD(K)
      X2 = XORD(L)
      Y1 = YORD(K)
      Y2 = YORD(L)
      XD = X2 - X1
      YD = Y2 - Y1
      T0 = ABS(XD).GE.ABS(YD)
      T1 = .NOT.T0
      T2 = X1.LT.X2
      T3 = Y1.LT.Y2
      IF (T0.AND.T2.OR.T1.AND.T3)  GO TO 120
      TEMP = X1
      X1 = X2
      X2 = TEMP
      TEMP = Y1
      Y1 = Y2
      Y2 = TEMP
  120 IF (T1)  GO TO 140
      N1 = (X1-XS)/DX
      N2 = (X2-XS)/DX
      DO 130  NX= N1,N2
      X = FLOAT(NX)*DX + XR
```

```
      Y = Y1 + YD*(X-X1)/XD
      NY = (Y-YS)/DY
      NY = NCOL1 - NY
  130 P(NX,NY) = ASTRK
      GO TO 160
  140 N1 = (Y1-YS)/DY
      N2 = (Y2-YS)/DY
      DO 150  NY = N1,N2
      Y = FLOAT(NY)*DY + YR
      X = X1 + XD*(Y-Y1)/YD
      NX = (X-XS)/DX
      NYY = NCOL1 - NY
  150 P(NX,NYY) = ASTRK
  160 CONTINUE
C
C     INTERNAL CONTOUR LINES
C
      NSC = 1
      NES = NELSKP(1)
      DO 420  N = 1,NUMEL
      IF (N.NE.NES)  GO TO 170
      NSC = NSC + 1
      NES = NELSKP(NSC)
      GO TO 420
  170 IF (NP(N,7).LE.0) GO TO 180
      NUMT = 4
      NPT(3) = NUMNP + N
      GO TO 190
  180 NUMT = 1
      NPT(3) = NP(N,3)
  190 DO 400  II = 1,NUMT
      JJ = IPERM4(II)
      NPT(1) = NP(N,II)
      NPT(2) = NP(N,JJ)
C
C     SORT FUNCTION VALUES
C
      DO 200 I = 1,3
      J = NPT(I)
  200 S(I) = F(J)
      NP1 = NPT(1)
      L = 1
      S1 = S(1)
      DO 220  I = 2,3
      IF(S(I).GE.S1)  GO TO 220
      NP1 = NPT(I)
      S1 = S(I)
```

```
      L = I
  22C CONTINUE
      L1 = NR(1,L)
      L2 = NR(2,L)
      IF (S(L1).GT.S(L2)) GC TO 240
      NP2 = NPT(L1)
      NP3 = NPT(L2)
      GO TO 250
  24C NP2 = NPT(L2)
      NP3 = NPT(L1)
  25C S2 = F(NP2)
      S3 = F(NP3)
      IF (S1.GT.FMAX.CR.S3.LT.-FMAX)  GO TO 400
      I = (S1+FMAX)/SPACE
      M = I - 9
      VALUE = FLOAT(M)*SPACE
      DIF13 = S3 - S1
      IF (DIF13.EC.C.)  GO TO 400
  26C IF (VALUE.GT.S3)  GC TC 40C
      NF = IABS(M) + 1
      IF (NF.GT.11)  GC TC 400
C
C     FINC END CCCRCINATES CF CONTOUR LINE SEGMENT
C
      XF = (VALUE-S1)/DIF13
       X = XORD(NF1)
      Y = YORD(NP1)
      X1 = X + XF*(XORD(NP3)-X)
      Y1 = Y + XF*(YCRD(NP3)-Y)
      NTT = NP1
      IF (VALUE.GT.S2)  NTT = NP3
      ST = F(NTT)
      DIFT2 = ST - S2
      IF (ABS(DIFT2).LT.1.E-8)  GC TO 350
      XF = (VALUE-S2)/DIFT2
      X = XORD(NP2)
      Y = YORD(NP2)
      X2 = X + XF*(XORD(NTT)-X)
      Y2 = Y + XF*(YORD(NTT)-Y)
      XD = X2 - X1
      YD = Y2 - Y1
      TO = ABS(XD).GE.ABS(YD)
      T1 = .NCT.TC
      T2 = X1.LT.X2
      T3 = Y1.LT.Y2
      IF (TO.AND.T2.CR.T1.AND.T3)  GO TC 270
      TEMP = X1
```

```
      X1 = X2
      X2 = TEMP
      TEMP = Y1
      Y1 = Y2
      Y2 = TEMP
C
C     STORE SIGNAL INTO P ARRAY
C
  270 IF (T1)  GC TC 300
      IF (XD.EQ.0.) GC TO 350
      N1 = (X1-XS)/CX
      N2 = (X2-XS)/CX
      DO 280  NX = N1,N2
      X = FLOAT(NX)*DX + XR
      Y = Y1 + YD*(X-X1)/XD
      NY = (Y-YS)/DY
      NY = NCCL1 - NY
  280 P(NX,NY) = XLAB(NF)
      GO TO 350
  300 N1 = (Y1-YS)/DY
      N2 = (Y2-YS)/DY
      DO 320  NY = N1,N2
      Y = FLOAT(NY)*DY + YR
      X = X1 + XD*(Y-Y1)/YD
      NX = (X-XS)/DX
      NYY = NCOL1 - NY
  320 P(NX,NYY) = XLAB(NF)
  350 M = M + 1
      VALUE = VALUE + SPACE
      GO TO 260
  400 CONTINUE
  420 CONTINUE
C
C     PRINT GRAPH
C
      PRINT 10, GRHEAC(1,NG),GRHEAD(2,NG),GRHEAD(3,NG)
   10 FORMAT (1H1, 50X, 3A4)
      NULL = 0
      PRINT 12, NULL, (I, I=10,100,10)
   12 FORMAT (1H0,I7,10I10 /1X)
      DO 450  J = 1,NCP
      L = J - 1
  450 PRINT 14, L, (P(I,J), I=1,NRP), L
   14 FORMAT (1H ,I4,2X,101A1,I4)
      PRINT 12, NULL, (I, I=10,100,10)
      PRINT 16
   16 FORMAT(//,T42,' CONTOUR LINE REFERENCE VALUES',/)
```

```
      DO 50C  I = 1,11
      W = FLOAT(I-1)*SPACE
      WN = - W
      IF (25-(((1CC*I)/2)-((I/2)*10C))) 460, 460, 470
 460  WRITE(6,18) XLAB(I), W, WN
      GO TO 500
 47C  WRITE(6,19) XLAB(I), W, WN
 50C  CONTINUE
  18  FORMAT(1H ,T11,A1,' = ',1P1E14.5,'  OR ',1P1E13.5)
  19  FORMAT(1H+,T68,A1,' = ',1F1E14.5,'  OR ', 1P1E13.5)
      WRITE(6,20)
  20  FORMAT(1H+,T73,'(*-DENOTES BOUNDRY POINTS)')
      GO TO 105
 600  RETURN
      END
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library Naval Postgraduate School Monterey, California 93940	2
3.	Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	2
4.	Naval Ship Systems Command (Code 2052) Navy Department Washington, D. C. 20360	1
5.	Professor Gilles Cantin Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	10
6.	LT John P. Malone, USN U. S. Navy Ship Repair Facility Box 34 FPO, San Francisco, California 96651	4
7.	Computer Facility Naval Postgraduate School Monterey, California 93940	1
8.	Professor J. E. Brock Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
9.	Professor P. F. Pucci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
10.	Mr. William Welsh Manager, Stress Group Marine Engineering The Westinghouse Electric Corp. Hendy Ave., Sunnyvale, California 94088	1
11.	LCDR Sompong Phasook, Royal Thai Navy 90 Soi Baring (Bangna) Sukumvid Road Bangkok, Thailand	1

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3 REPORT TITLE

A Computer Program for the Analysis of Linearly Elastic Plane-Stress, Plane-Strain Problems

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

None

5. AUTHOR(S) (First name, middle initial, last name)

John Patrick Malone

6 REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1968	132	16

8a. CONTRACT OR GRANT NO. NA b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) NA 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10 DISTRIBUTION STATEMENT

This document is subject to special export controls and each transmittal to foreign government or foreign nationals may be made only with prior approval of the Superintendent, Naval Postgraduate School, Monterey, California 93940

11 SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13 ABSTRACT

The computer program for the analysis of linearly elastic plane-stress or plane-strain problems devised by Felippa in his work on "Refined Finite Element Analysis of Linear and Nonlinear Two-dimensional Structures" has been modified to include the use of initial displacement boundary conditions. In addition the original IBM 7094 computer dependent program has been adapted for use on the IBM 360/65 computer. In both programs the FORTRAN IV language has been used.

Problems involving "Poor fit" displacement boundary conditions and refined mesh analysis using coarse mesh analysis input displacements, which could not have been done with the original program, are now possible with the modified version presented herein.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified

Security Classification

A-31408

14 KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Finite Element Computer Program						
Plane-Stress or Plain-Strain Computer Analysis						

DD FORM 1 NOV 65 1473 (BACK)

S/N 0101-807-6821